ALZHEIMER'S DISEASE BIOMARKER DISCOVERY AND DATA VISUALIZATION USING PROTEOMICS AND BIOINFORMATICS APPROACHES

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Mark David D'Ascenzo

May 2013

ABSTRACT

Neurological disorders such as Alzheimer's disease and Parkinson's disease possess complex pathologies that are only partially understood. As more comprehensive and sophisticated studies are implemented in an effort to further understand the underlying pathologies of such disorders, the generation of larger and more complex quantitative output is becoming increasingly more commonplace. Extracting relevant biological insight from such output can be challenging and often requires the application of sophisticated computational tools that are capable of reducing complexity so that potentially biologically relevant patterns can emerge. In this thesis, two machine learning classification algorithms, Linear Discriminant Analysis and Random Forests™, are applied to a complex proteomics data set derived from a multi-subject study of human Cerebral Spinal Fluid (CSF) using a 2d-gel electrophoresis in an effort to identify novel Alzheimer's disease and Parkinson's disease biomarkers and the results are reported. Additionally, a review of recent proteomic studies focused on the discovery of novel Alzheimer's disease biomarkers within CSF is presented. Described also is a novel visualization tool *iTRAQPak*, which was successfully applied to the analysis of CSF based iTRAQ™ protein expression data sets obtained from a cohort of Alzheimer's disease subjects participating in a Phase I drug trial.

BIOGRAPHICAL SKETCH

Mark D'Ascenzo received his B.S. degree in Biology from the Pennsylvania State University Eberly College of Science. After completion of his undergraduate studies, he worked professionally in an academic research setting for a number of years, where he gained expertise in molecular and cellular biology and bioinformatics based research methods. In his most recent position prior to his graduate training, he worked as a Bioinformatics Specialist at the Boyce Thompson Institute for Plant Research at Cornell University. Here he was exposed to a number of genomics based research projects relating to the sequencing and genome-wide analysis of the bacteria *Pseudomonas syringae*. With an interest in diversifying his background and gaining exposure to the field of proteomics, he pursued graduate training in the Department of Biomedical Engineering at Cornell University under the guidance of Prof. Kelvin Lee. Working under Prof. Lee, he addressed pertinent topics in proteomics based research of Alzheimer's disease. Topics addressed in his graduate training include Alzheimer's disease biomarker research using proteomics and bioinformatics based methods and the visualization and analysis of complex proteomics data sets. Mark is currently employed as Bioinformatics Scientist at Roche NimbleGen, Inc. in Madison, WI.

This thesis is dedicated to my family and to the friends who have offered support throughout my graduate studies.

ACKNOWLEDGMENTS

I would like to thank Prof. Kelvin Lee for his guidance and numerous contributions to the work presented in this thesis. I would also like to thank Erin Finehout for her contributions to the 2D gel analyses required for the generation of data sets used in the analyses described in Chapter 2, and also for helpful thoughts on Random Forest analysis. I would also like to thank Leila Choe for her numerous contributions, especially her work with the iTRAQ experimental procedure which was essential to work presented in Chapter 3. And finally, I would like to acknowledge Dr. Norman Relkin for his contributions to the review presented in Chapter 1 and for CSF samples used in the work presented in Chapter 3.

PREFACE

This thesis consists of four chapters addressing Alzheimer's disease and proteomics based research at several levels. The first three of these chapters are outlined below. The remaining chapter offers concluding remarks pertaining to work presented in the first three chapters and also offers possible avenues for future research.

A review article is presented in Chapter 1, entitled *Alzheimer's disease cerebrospinal fluid biomarker discovery: A proteomics approach*, which was published in the December 2005 issue of *Current Opinion in Molecular Therapeutics*. This review article contains a comprehensive overview of relevant and timely research methods, literature, and research results pertaining to the characterization and identification of novel CSF proteins that may someday allow accurate diagnosis of Alzheimer's disease in living individuals. While encompassing research articles prior to its publication in 2005, this review continues to be actively cited and remains a useful guide to AD biomarker research methods and literature.

The second chapter entitled: *Biomarker discovery and classification of proteomics data using two different classification algorithms*, describes the analysis of several proteomics data sets using two prominent classification algorithms. Here, two classification algorithms are introduced: Linear Discriminant Analysis and Random Forest. Both algorithms are then applied to several proteomics data sets in an effort to identify groups of proteins that allow the algorithms to accurately classify the samples contained within the data sets. In these analyses, samples were obtained from healthy individuals, patients with Alzheimer's disease, or patients with Parkinson's disease. Proteins identified as being important by the algorithms are then used to blindly classify subjects and are assessed for their ability to accurately classify the samples contained within the data sets.

In the third chapter, the reader is presented with another previously published work entitled: *iTRAQPak: an R based analysis and visualization package for 8-plex isobaric protein expression data*. This work was published in *Briefings in Functional Genomics and Proteomics*

in February, 2008. iTRAQ (trademarked by Applied Biosystems) is a mass spectrometry-based technology which enables the quantitative analysis of protein expression using a multiplex approach that allows up to 8 samples (8-plex iTRAQ) to be analyzed in a single experiment. The often large data sets derived from just a single iTRAQ based experiment can be quite complex and a challenge to analyze. Using the R statistical and visualization environment, the iTRAQ package, or iTRAQPak, was developed to help facilitate the visualization and analysis of iTRAQ based expression data. The package offers a number of features to facilitate data analysis, including sample normalization, scaling, and plotting methods. Among the more valuable features offered by the package is the expression plotting function, which offers a birds-eye view of protein expression patterns within the data using a number of novel visualization approaches. The utility of this package is demonstrated through its application to the analysis of 8-plex iTRAQ protein expression data obtained from cerebrospinal fluid samples from Alzheimer's disease subjects involved in a Phase I drug trial

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2DGE	2 dimensional gel electrophoresis
AD	Alzheimer's disease
CSF	cerebrospinal fluid
GSFLD	Fisher's Linear Discriminant Analysis gene shaving algorithm
HPLC	high pressure liquid chromatography
ICAT	isotope-coded affinity tag
iTRAQ	isobaric tags for relative an absolute quantitation
LDA	Linear Discriminant Analysis
N	Normal subject, individual undiagnosed with disorder
PCA	Principal Components Analysis
PD	Parkinson's disease
RF	Random Forest algorithm
RFShave	Random Forest shaving algorithm

CHAPTER 1*

ALZHEIMER'S DISEASE CEREBROSPINAL FLUID BIOMARKER DISCOVERY: A PROTEOMICS APPROACH

1.1 Introduction

The ability to accurately and definitely diagnose the presence of Alzheimer's disease (AD) in living individuals would have a significant impact on the treatment of this disease. A clinical diagnosis can be made with a reasonable degree of accuracy by experienced neurologists; however, the identification of reliable biomarkers for AD would significantly improve the assessment of the disease and may lead to new targets for intervention, in addition to the ability to assess disease state and responses to available treatments. Towards this end, there has been significant interest in monitoring the changes in cerebrospinal fluid (CSF) protein expression that may result from AD pathology. CSF protein expression may be more representative of changes occurring in the brain than changes in serum protein expression, and lumbar puncture is a less invasive procedure than brain biopsy for obtaining tissue. CSF contains approximately 2000 proteins and peptides, and the ability to quantify changes in expression of all of these molecules, and to then compare these expression patterns across many individuals is not trivial. One of the reasons for the difficulty identifying biomarkers through this approach is rooted in the complex biochemistry of the disease. However, an equally important limitation relates to the available technologies for profiling protein expression (also known as 'proteomics'). There is no single technology capable of reliably monitoring all proteins that are expressed by living systems because of the diversity of physical and chemical properties of proteins. Furthermore, there is a need for improved algorithms for analyzing data obtained from large-scale proteomics

---

* Chapter 1 was published in *Current Opinion in Molecular Therapeutics* (2005) 7(6):557-64.

* Chapter 3 was published in *Briefings in Functional Genomics and Proteomics* (2008) 7(2):127-35.

experiments such as those that are involved in biomarker discovery. Nonetheless, there are several commonly used methods and technologies that show tremendous promise.

In this review, recent published studies that attempt to identify AD biomarkers within the CSF proteome are considered. Generally, the technologies that are employed in this literature fall into two categories: gel-based separations followed by mass spectrometry (MS) or liquid-based separations followed by MS. The various technologies that are being implemented will briefly be discussed, and emerging proteomic technologies and approaches being applied to AD CSF biomarker studies from recent published literature will be focused on. None of the approaches considered here has gained widespread acceptance by the clinical community, nonetheless, these methods show promise and highlight both the advantages and limitations of using proteomics for biomarker discovery.

1.2 AD Biomarkers

AD is the most common form of neurodegenerative dementia among the elderly, is estimated to affect between 6 and 10% of the population over the age of 65, and is fatal with no widely accepted treatment available to slow or reverse its progression [1]. It is estimated that in the US, AD treatment costs range from US $50 billion to $100 billion annually, making it the third most-costly disease after cardiovascular disease and cancer [1]. There is a desire to better understand the molecular pathology of AD, as available diagnostic approaches are not optimal to detect the onset, presence or progression of this disease, or to assess the effectiveness of new clinical treatments [2]. Clinical diagnosis of probable AD is typically derived through patient evaluation or cognitive testing. Patient evaluation may involve the use of established criteria set forth in the Diagnostic and Statistical Manual of Mental Disorders (DSM) [3], and by the National Institute of Neurological and Communicative Diseases and Stroke/Alzheimer's Disease and Related Disorders Association (NINCDS-ADRDA) [4]. Sensitivity of these criteria for AD diagnosis ranges from 76to 98% and specificity ranges from 61 to 84% [5]. Cognitive tests may include the

use of computerized examination tests or paper-based neuropsychological batteries. However, inherent biases in these tests, such as patient variation in cognitive skills, socioeconomic background, poor test-taking skills and test anxiety may lead to misdiagnosis [5]. The complementary use of computerized tomography, magnetic resonance imaging and positron emission tomography scanning technology within a clinical setting is still being established, and the efficacies of these technologies for AD diagnosis have been reported with varying ranges of sensitivity and specificity measures [6]. The primary challenge is that a definitive diagnosis for AD requires a postmortem examination; thus an assessment of probable AD is the best that can be routinely achieved by clinical diagnosis.

This issue is particularly important to consider. The output from any biomarker study is a function of the individuals that are studied and the control groups that are used. If the AD group contains individuals that have no postmortem confirmation, then a special emphasis must be placed on validating those observations in a group with a definitive diagnosis. Furthermore, if the control group contains only individuals that have no known neurological disorder at the time of sampling, then the observed biomarkers may not be effective in a differential diagnosis. In AD biomarker studies, an idealized population would involve premortem CSF samples from AD patients with postmortem confirmation and a collection of normal and neurological controls to aid in the identification of AD-specific biomarkers. Phenotypic overlap can blur the boundaries of differential diagnosis and is likely to confound the identification of disease-specific biomarkers by proteomic analysis. AD-like neuropathology often exists in combination with that of other disorders, such as Lewy bodies and cerebrovascular disease. This mixture of pathologies adversely impacts on the accuracy of clinical differential diagnosis, and highlights the importance of obtaining autopsy correlation. Ultimately, the proteome may prove to be an excellent means of deconvolving mixed phenotypes ante mortem, but only if biomarkers for the individual disorders can be identified first.

Although these approaches offer an exciting opportunity to discover new biomarkers, they also highlight an important challenge in the field. The identification of CSF AD biomarkers

involves identifying biomarkers in clinically affected individuals (i.e., antemortem CSF) with postmortem confirmation of the disease (i.e., definite diagnosis). Clearly, proteomic studies that rely on samples with no postmortem confirmation are of limited use, as the sensitivity and specificity of any resulting biomarkers cannot be better than the best clinical diagnosis. The analysis of postmortem CSF (such as that which may be obtained upon postmortem examination) is certainly not representative of antemortem CSF protein expression because of the many biochemical changes that occur upon death (e.g., blood-brain barrier function). Thus, the ideal study would involve samples collected antemortem from individuals for which postmortem confirmation is available. Furthermore, the ability to diagnose a healthy individual from one with AD is relatively straightforward, so the more interesting studies involve the differential diagnosis of AD from other dementias. Finally, the study should have a large enough number of samples to be statistically meaningful.

The discovery of disease biomarkers that would allow rapid and non-invasive assessment of disease state may assist to alleviate deficiencies in current diagnostic methods, possibly allow early detection of the disease, provide a means to monitor progression, and may reduce research and development costs of new treatments, in addition to the time required to conduct clinical trials. Biomarkers are historically termed as 'analytes in biological samples, any measurement that predicts a person's disease state, or response to a drug', but has evolved to include 'imaging modalities or multi-marker genomic/proteomic panels' [7]. Ideally, a validated panel of AD biomarkers would allow an unambiguous determination of disease state. Some of the best known putative AD biomarkers include CSF and Aβ42; however, these markers can only contribute to a clinical diagnosis and are not accepted as a surrogate for definitive diagnosis. As a result, there is interest in the discovery of new protein-based biomarkers using CSF. In this review, technologies and studies targeting new CSF biomarkers will be considered and the reader will be directed to other important literature assessing well known CSF biomarkers [8].

1.3 CSF

CSF transports cellular products, metabolites, neurotransmitters and proteolytic fragments [9], and is secreted from several central nervous system (CNS) structures, including the choroid plexus. A highly regulated blood-brain barrier separates CSF from the peripheral circulatory system, and the proteins that are found in the CSF are believed to come either from the brain (~ 20%) or from blood (~ 80%) [10]. The CSF proteome predominately consists of serum albumin, transferrin and immunoglobulin(Ig) isoforms and these represent > 70% of the total proteinin CSF [11]. Although the proteins found in CSF derive from multiple sources, the brain-derived proteins are generally considered to more likely be relevant in AD biomarker studies. It is easy to appreciate that biochemical changes in the brain may lead to changes in CSF protein expression; however, it is equally important to recognize that CSF biomarkers might also represent important changes in protein expression in the plasma. As a result, it is critical to consider the complex dynamics of CSF flow and of changes in protein expression. A major challenge associated with the diverse sources of CSF proteins is that there is a very large dynamic range of CSF protein expression. Indeed, the diversity of protein expression is estimated to be 12 orders of magnitude [12]. This challenge is important because the dynamic range of CSF protein concentrations exceeds the dynamic range of proteomic detection technologies, making it especially difficult to accurately and consistently identify and quantify CSF proteome constituents using currently available methods.

1.4 Immunodepletion and Prefractionation in Sample Preparation

One common approach to address the issue of the wide dynamic range of CSF protein expression is to employ immunodepletion methods to remove the predominant proteins, and there are a variety of commercially available kits for carrying out this step. The main benefit to immunodepleting these abundant CSF proteins is that the analytical techniques used to study CSF can emphasize proteins of lower concentration in a separate experiment from the more

abundant proteins. Because of this potential benefit, the use of immunodepletion methods is widespread among CSF literature [12,13-15]. The Affinity Removal System immunoaffinity column (Agilent Technologies) was used by Maccarrone *et al*. to remove human serum albumin (HSA), transferrin, haptoglobin, IgG, IgA and antitrypsin from CSF samples [12,13]. Additionally, Maccarrone *et al*. observed that immunodepleted samples separated on sodium dodecyl sulfate polyacrylamide gel electrophoresis (SDS-PAGE) and analyzed by MS enabled the identification of 240 less-abundant proteins, and immunodepleted samples sub-fractionated by anion exchange chromatography yielded the identification of 112 proteins, while non-depleted and non-fractionated CSF samples identified only 38 proteins [12,13]. Depletion methods used by Ogata *et al*. [15] allowed 50 proteins to be identified, of which all but three were detected in the Maccarrone *et al*. study [12,13], however six proteins were previously unannotated on 2-dimensional gels [15]. An important consideration in the use of these methods is that there may be a loss of non-targeted proteins through non-specific binding [16]. There may also be a loss of information related to the possible utility of certain cleavage products of the predominant proteins as biomarkers.

Prefractionation methods can be used to distribute sample complexity into multiple, less complex samples, prior to analytical analysis and this is a more general approach to simplifying mixtures of proteins than immunodepletion. In a study by Yuan and Desidario, a prefractionation technique was developed in which CSF samples were separated using a reversed-phase solid-phase extraction (SPE) cartridge, further partitioned and selectively extracted by organic solvents, and finally separated by 2-dimensional protein gel electrophoresis (2DE) [17]. This procedure resulted in the identification of 42 proteins, many of which were of very low abundance, and 46% of which were not identified by the Maccarrone *et al*. study [12,13]. The development of novel and effective strategies for reducing sample complexity and addressing the dynamic range issue are technical limitations that need to be overcome before the full benefit of proteomics in CSF biomarker discovery can be achieved.

1.5 General Proteomic Methods

There are two basic strategies for CSF protein biomarker studies: gel- and liquid-based separations. Both of these approaches are often followed by some type of MS. The use of 'upstream separations' (either gel or liquid) is required because the mass spectrometers currently available are unable to simultaneously analyze more than a few proteins in a single analysis at any given time. The resolution of the mixture of approximately 2000 proteins and peptides in CSF into simpler fractions that can be studied by the mass spectrometer individually, greatly improves the likelihood of identifying meaningful changes in protein expression. A mass spectrometer is an instrument that measures the mass of molecules with high accuracy and includes two key steps, the ionization of the analyte into charged gas phase ions and the determination of the mass to charge ratio of those ions. A mass spectrometer is typically used for the final analysis of CSF proteins because it can uniquely identify an unknown protein based on a computer analysis of mass spectra compared to an available sequence database. For example, an unknown protein that has been resolved by gels or liquid separations can be identified in terms of the underlying gene as well as many post-translational modifications that have occurred to that unknown protein using a mass spectrometer. Furthermore, in certain experiments, the mass spectrometer can be used to quantify the change in expression level of a given protein across multiple samples, although most of the biomarker studies to date rely on quantification using gel-based separations or other methods as described below.

1.6 Biomarker Studies Using Gel-Based Separations

In 2DE, the proteins from a given sample are collected and separated first by isoelectric point and then by size using the techniques of isoelectric focusing (IEF) and SDS-PAGE. Since charge and size are independent characteristics for a given protein, this gel-based technique (originally developed in 1975 [18]) has unsurpassed resolution compared with other available methods. Anecdotal reports include the ability to resolve complex mixtures of proteins into as many as

10,000 features on a gel. After separation by SDS-PAGE, proteins are fixed into the gel and are often stained with a total protein stain, an example of which is shown in Figure 1.1. The underlying assumption is that the stain intensity of the spots that appear on the gel relates to the quantity of the protein present. Thus, by comparing changes in spot intensity across multiple gels derived from the CSF proteins from multiple individuals, one might be able to identify changes in protein expression that relate to a given disease state. The process by which images are compared is facilitated by several commercially available software packages. This general approach has some important technical limitations that need to be considered. Firstly, not all proteins can be resolved well by 2DE; in particular, small, large, very acidic, very basic and many hydrophobic proteins are not resolved clearly on 2DEimages [19]. Secondly, the 2DE procedure is extremely laborious, time-consuming and not amenable to automation. For example, only a few samples can be run per week by an experienced technician. Furthermore, because of the complexity of the protocol, there are many opportunities to introduce errors in the experiment, leading to somewhat limited laboratory-to-laboratory reproducibility (undesirable in the context of biomarkers). Thirdly, the available software packages for image analysis cannot robustly handle certain aspects of the resulting images, including technical artifacts and certain posttranslationally modified protein forms. As a result, a very careful image analysis of many gel images may take many days or weeks to complete and requires significant labor. Nonetheless, this 2DE approach has been commonly used to build CSF protein maps [20,21] and in biomarkers studies, because (i) unlike other methods, it is scalable across a large number of samples, (ii) it can be used to study changes in the post-translationally modified forms of proteins, and (iii) there is an established record of the use of this approach in identifying biomarkers for neurodegenerative diseases [20,22-24]. Indeed, the technologies for 2DE have evolved to the point where the method is sensitive enough to identify proteins that have never previously been reported to occur in CSF [21].

Figure 1.1 A representative 2DE image of CSF from an AD patient

Proteins are separated by charge in the horizontal direction and by size in the vertical direction. Proteins have been visualized with silver stain. A comparison of such 2DE images from a number of different individuals can be used to study cerebrospinal fluid (CSF) Alzheimer's disease biomarkers. The change in stain intensity can be used as a measure of the amount of protein expressed in the CSF of that particular individual.

The outcome of the 2DE analysis of CSF proteins is an image and a series of spots on the image that change in stain intensity across samples of interest. By physically isolating those spots of interest in the gel and carrying out a sequence-specific protease digestion of the spots, a mass spectrometer can then be used to determine the masses of the resulting peptides, in addition to amino acid sequence-specific information.

As mentioned previously, a key step in the analysis of molecules by a mass spectrometer involves the ionization of those molecules prior to mass analysis. The ionization of proteins and peptides commonly involves one of two methods: matrix assisted laser desorption/ionization (MALDI) or electrospray ionization. Both of these methods work well for the analysis of proteins, but neither works well for all types of proteins. Thus, many laboratories have a suite of mass spectrometers capable of both ionization types. Several studies report the use of 2DE to study CSF for AD biomarker discovery [20,25,26]. In the more recent of these studies, Puchades *et al*. compared the CSF proteome of probable AD (n = 7) and non-AD (n = 7) disease patients using 2DE-MS to identify altered levels of nine proteins: apolipoprotein A1, apolipoprotein E, apolipoprotein J, β-trace, retinol-binding protein, kininogen, α1-antitrypsin, cell cycle progression 8 protein and α-1β glycoprotein [26]. Ogata *et al*. used 2DE-separated CSF proteins to explore strategies to stain for phosphoproteins and glycoproteins [15]; such methods may prove useful to post-translational modification (PTM) studies of neuropathology in AD and biomarker discovery. The role that PTMs may play in AD remains unclear, however two 2DE-based studies offer methods that may allow further assessment of their relevance [27,28].

1.7 Biomarker Studies Using Liquid-Based Separations

The other approach to the study of the CSF proteome that has been used in the literature involves liquid-phase separations rather than gel-based separations. These methods may include a single separation step or multiple steps and are followed by MS analysis.

1.7.1 Capillary Electrophoresis-MS

Using thin-walled columns typically coated with a charged material such as silica, capillary electrophoresis (CE) is a separation method that can be easily automated and coupled 'on-line' to some mass spectrometers. That is, the column eluent can flow directly into a mass spectrometer using electrospray ionization. Although CE alone does not offer the same resolution that is

available by 2DE, the separations can be carried out significantly faster [21,29]. Because CE is simpler than 2DE and can be automated, the separation schemes are far more reproducible than other methods [21,29]. While CE-MS has been used in a variety of biomarker studies [29,30,31], there is only one recent example of the application of this method to AD CSF studies. Wittke *et al*. compared CSF from healthy volunteers (n = 4) and probable AD patients (n = 8), which resulted in the identification of four differentially expressed polypeptides that were 13.4, 11.78, 11.98 and 4.82 kDa in size [29]. Their technique was especially useful in resolving polypeptides in the low-molecular weight (MW) range (0.8to 15 kDa), in which 450 polypeptides were identified. In a comparable 2DE-based study of CSF, approximately 50 protein spots in the 0.8- to 15-kDa range were identified, but many more (~ 550 protein spots) were identified in the 15- to 150-kDa MW range [21]. While the Wittke *et al*. study did not use well characterized CSF samples and did not further characterize the differentially expressed polypeptides identified, the investigators demonstrated the important proof-of-principle application of CE-MS technology to the study of CSF [29].

1.7.2 Shotgun Proteomics

A primary issue with the CE-MS approach is the limited resolution offered by CE alone. As a result, there is significant effort underway to employ 'shotgun proteomics' methods. These shotgun experiments usually involve two liquid-phase separations upstream of MS. The most common implementation of shotgun proteomics includes a strong cation exchange separation (where many fractions might be collected) followed by a reverse-phase high performance liquid chromatography (HPLC) experiment for each cation exchange fraction. The eluent from each of the HPLC experiments can either be coupled on-line directly into the electrospray ionization source of a mass spectrometer, or can be used off-line by applying the column eluent as nanoliter-sized fractions onto a MALDI target plate (this approach is often called liquid chromatography (LC)-MALDI). Because these methods rely on multiple dimensions of LC prior

to MS, they are often referred to as 2DLC or MDLC experiments. A key feature of this approach is that the entire protein content of the sample of interest (e.g., CSF) is initially digested into a highly complex mixture of peptides (e.g., 100,000 tryptic peptides) using a sequence-specific protease. The resulting mixture of peptides is then separated and studied by MS and the entire set of MS spectra are compared against available sequence databases. A key limitation of the shotgun approach as applied to biomarker discovery is that these methods are not easily scaled for studies involving a large number of samples and there are at least two reasons for this. Firstly, a relatively large amount of time is required to analyze a small number of samples with this approach because the data sets are large, the algorithms for assigning mass spectra are not yet ideal, and the organization of the resulting data can be complex. Secondly, the technique does not acquire information on the same set of proteins from experiment to experiment.

Although there may be overlap in the proteins studied from one experiment to the next over a large number of shotgun analyses, the number of proteins studied in all experiments is small. Indeed, the problem is compounded when experiments are carried out in different laboratories and using different types of mass spectrometers, and a recent study by six companies using 10,000 human cells identified total of 1757 proteins, but only 3% of these proteins were identified by all of the groups [32]. On the other hand, these shotgun methods offer an ability to measure a broad group of proteins, including hydrophobic proteins and proteins present in relatively lower abundance - two classes of proteins that are not well studied by 2DE. The early efforts at shotgun proteomics are typically used if only as a descriptive method to determine what proteins may be present in the sample. However, the technique is somewhat limited because the absence of a particular protein from an analysis does not mean that the protein is absent from the sample, it only means that it was not identified during that particular experiment. As mentioned previously, the number of proteins observed over a large number of experiments can be relatively small. To improve the utility of the shotgun approach, there has been an effort to use isotopic or isobaric tags to study multiple samples at once and to determine the expression ratio among the samples studied for the proteins observed in the experiment. We provide

examples of both of these types of analyses below. For a detailed discussion of the various types of mass spectrometers and the fundamentals of shotgun proteomics methods, see references [33,34]. Wenner *et al*. analyzed CSF from healthy, aged individuals using shotgun proteomics [14]. They used the standard approach of a strong cation exchange separation and reversed-phase separation coupled to quadropole ion trap MS to try to identify as many proteins as possible from CSF. The investigators identified 249 CSF proteins from ten individuals using this strategy. The results provide a useful measure of the proteins that may be found in CSF and demonstrate the applicability of the shotgun method to study CSF proteins.

The shotgun approach can quantify changes in protein expression by including either isotopic or isobaric tags; details of these methods can be found in references [35,36].Two of the most commonly used methods include the isotope-coded affinity tag (ICAT) and isobaric tags for relative and absolute quantitation (iTRAQ) reagents. Both of these sets of reagents are used to differentially label proteins or peptides derived from different samples. After labeling, the proteins or peptides from different samples are combined together and analyzed as a single sample using 2DLC and MS. Data analysis relies on the differential labels to distinguish the origin of the samples and to establish the expression ratio of the proteins observed in the experiment. In the case of ICAT, two samples can be studied in a single experiment, whereas with the iTRAQ reagents, up to four samples can be compared in a single experiment. The ICAT method can label either peptides or intact proteins, whereas the iTRAQ method can only label peptides. Two recent examples of the application of ICAT to CSF studies are cited below. The iTRAQ technique is relatively new and there are no reports of the application of this method to biomarkers studies. Thus, these examples suggest the possible impact of iTRAQ on biomarker studies.

The use of ICAT to quantify differences in CSF protein expression among elderly patients was explored by Zhang *et al*. [37,38]. In one study, age-related changes in CSF were investigated using ICAT [37] and, in the other, ICAT was used to quantify differences in CSF between probable AD patients (n = 32) and age-matched controls (n = 31) [38]. The latter study

resulted in the identification of 390 CSF proteins, where concentration ratios were calculated for 42% of these, and differential expression ratios of 20% were found in over half of the proteins [38]. The investigators compared their findings with two other CSF AD biomarker studies [26,39] and reported that α2-microglobulin was the only consistently increased protein in all of these studies. It is important to recognize that each of these studies used complementary and different types of analyses to achieve this result [26,37,38,39]. An important feature of the Zhang study is the use of an alternate technique, Western analysis, to validate these findings [38]. Although iTRAQ technology [35] has not yet been applied to the study of CSF, it was used to study changes in protein expression from Escherichia coli cultures [40]. Using this technique, the expression ratios of > 780 proteins could be quantified in a single experiment and included the measurement of low abundance proteins [40]. A comparison of the iTRAQ data to 2DE data from the same samples suggested that the coefficient of variation in the iTRAQ measurements (0.24) was better than that achieved using 2DE (0.31) [41]. For studies that involve the quantification of expression ratios using shotgun proteomics, the iTRAQ method should offer better throughput and reliability than other methods. However, the applicability of this approach for studies involving a substantial number of patient samples is not clear at this time.

Biomarker studies using other methods Surface-enhanced laser/desorption ionization (SELDI)-MS is an alternate approach for the MS-based identification of biomarkers. Unlike the gel- and liquid-based separation methods discussed above, the SELDI technology relies on surfaces that have chromatographic properties. CSF proteins can be bound to these surfaces based on various physicochemical properties. After attachment, the proteins can be interrogated using a time-of-flight mass spectrometer. The advantage of this technology is that there is no need for upstream separations so that the sample analysis time is extremely fast. However, the technology, as it has been implemented to date, has a few limitations. Firstly, the resolution and capabilities of the mass spectrometer are not near the level of the analytical instruments used in the gel-based or shotgun proteomics methods. The limited resolution means that proteins cannot always be easily and unambiguously identified with this approach. Secondly, the mass range that

is studied is small - typically less than a few thousand Daltons. Many proteins exist as larger molecules in CSF, and proteolytic degradation, whether by in vivo processes or in vitro storage, can occur on a short timescale. Thus, there are questions about the ability of this approach to study a representative and reproducible sampling of the CSF proteome. However, improvements in technology may begin to address these concerns in the near future. There are several reports demonstrating the use of SELDI-MS to study disease biomarkers derived from CSF. In 2003, Carrette *et al*. reported a study of CSF from probable AD patients (n = 9) where the strong anionic exchange surface was employed to detect five differentially expressed polypeptides compared with controls (n = 10) [39]. Four of these peptides were purified and identified as cystatin C, two β2-microglobulin isoforms, a4.8-kDa VGF polypeptide and an unnamed 7.7-kDapolypeptide. More recently, Lewczuk *et al*. identified three novel amyloid-β peptides using a biochemically treated SELDI chip array containing amyloid-specific antibodies [42]. Sanchez *et al*. demonstrated the use of SELDI-MS to identify a 13.4-kDa CSF protein (later characterized to be cystatin C) in a small group of patients (n = 8) with Creutzfeldt-Jakob disease using a copper affinity array [43].

1.8 Conclusion

There is great promise in the application of proteomic approaches to the discovery of CSF biomarkers for Alzheimer's disease and many other neurodegenerative diseases. There are several limitations in the current technology that somewhat hinder progress in the field to date. These include issues both with gel- and liquid-based separation strategies. As a result, it is important for investigators to use complementary techniques to validate any interesting results derived from these MS-based analyses rather than rely on the observations derived from a single method. A second important consideration is the need for improved computer algorithms related to these efforts. Improvements in image analysis algorithms for comparing 2DE data sets could significantly shorten the time required for analyzing the raw gel-based data. Improvements in

algorithms for matching MS spectra to sequence databases would help ensure that all of the high quality MS data obtained is used most effectively and analyzed quickly. Improvements in the availability and application of statistical methods for underspecified data sets (more ‘features' than 'samples') would help refine the identification of biomarkers (not discussed in this review). A third important issue is the limited availability of clinically well characterized samples for inclusion in any such studies.

Ideally, a large repository of antemortem CSF from a variety of individuals with AD and other neurological disorders could be provided to the community for use as a common reference set that can be studied using all of these techniques. However, there is limited availability of antemortem CSF samples with postmortem confirmation of the disease. As a result, many of the proteomics studies to date include the analysis of probable AD samples only or include only a very small number of definite AD samples, limiting the statistical significance of the results. Despite the current limitations in the field, there is tremendous potential and enthusiasm by the community in biomarker discovery. There are ongoing improvements in the technology across all of the methods described. These improvements and developments include increases in throughput, robustness, sensitivity and dynamic range of the technologies. Much of the technological development is motivated by the potential clinical impact of newly discovered CSF biomarkers and given the very intriguing preliminary data coming from these efforts; there will likely be interesting and useful discoveries in the near future.

REFERENCES

1. Leifer BP: **Early diagnosis of Alzheimer's disease**: Clinical and economic benefits. *J Am Geriatr Soc* (2003) 51(5 Suppl):S281-S288.

2. Ho L, Sharma N, Blackman L, Festa E, Reddy G, Pasinetti GM: **From proteomics to biomarker discovery in Alzheimer's disease**. *Brain Res Brain Res Rev* (2005) 48(2):360-369.

3. American Psychiatric Association: **Task Force on DSM-IV. In:** ***Diagnostic and Statistical Manual of Mental Disorders: DSM-IV***. American Psychiatric Association, Washington, DC, USA (2000).

4. McKhann G, Drachman D, Folstein M, Katzman R, Price D, Stadlan EM: **Clinical diagnosis of Alzheimer's disease: Report of the NINCDSADRDA Work Group under the auspices of Department of Health and Human Services Task Force on Alzheimer's Disease**. *Neurology* (1984) 34(7):939-944.

5. Zamrini E, De Santi S, Tolar M: **Imaging is superior to cognitive testing for early diagnosis of Alzheimer's disease**. *Neurobiol Aging* (2004) 25(5):685-691.

6. Frisoni GB: **Structural imaging in the clinical diagnosis of Alzheimer's disease: Problems and tools**. *J Neurol Neurosurg Psychiatry* (2001) 70(6):711-718.

7. Baker M: **In biomarkers we trust**? *Nat Biotechnol* (2005) 23(3):297- 304.

8. Andreasen N, Blennow K: **CSF biomarkers for mild cognitive impairment and early Alzheimer's disease.** *Clin Neurol Neurosurg* (2005) 107(3):165-173.

9. Romeo MJ, Espina V, Lowenthal M, Espina BH, Petricoin EF 3rd, Liotta LA: **CSF proteome: A protein repository for potential biomarker identification**. *Expert Rev Proteomics* (2005) 2(1):57-70.

10. Reiber H: **Dynamics of brain-derived proteins in cerebrospinal fluid**. *Clin Chim Acta* (2001) 310(2):173-186.

11. Sickmann A, Dormeyer W, Wortelkamp S, Woitalla D, Kuhn W, Meyer HE: **Towards a high resolution separation of human cerebrospinal fluid**. *J Chromatogr B Analyt Technol Biomed Life Sci* (2002) 771(1- 2):167-196.

12. Maccarrone G, Birg I, Malisch E, Rosenhagen MC, Ditzen C, Chakel JA, Mandel F, Reimann A, Doertbudak C-C, Haegler K, Holsboer F, Turck CW: **In-depth analysis of the human CSF proteome using protein prefractionation**. *Clinical Proteomics* (2004) 1(3-4):333-364.

13. Maccarrone G, Milfay D, Birg I, Rosenhagen M, Holsboer F, Grimm R, Bailey J, Zolotarjova N, Turck CW: **Mining the human cerebrospinal fluid proteome by immunodepletion and shotgun mass spectrometry**. *Electrophoresis* (2004) 25(14):2402-2412.

14. Wenner BR, Lovell MA, Lynn BC**: Proteomic analysis of human ventricular cerebrospinal fluid from neurologically normal, elderly subjects using two-dimensional LC-MS/MS**. *J Proteome Res* (2004) 3(1):97-103.

15. Ogata Y, Charlesworth MC, Muddiman DC: **Evaluation of protein depletion methods for the analysis of total-, phospho- and glycoproteins in lumbar cerebrospinal fluid**. *J Proteome Res* (2005) 4(3):837-845.

16. Raymackers J, Daniels A, De Brabandere V, Missiaen C, Dauwe M, Verhaert P, Vanmechelen E, Meheus L**: Identification of twodimensionally separated human cerebrospinal fluid proteins by Nterminal sequencing, matrix-assisted laser desorption/ionisation - mass spectrometry, nanoliquid chromatography-electrospray ionization-time of flight-mass spectrometry, and tandem mass spectrometry**. *Electrophoresis* (2000) 21(11):2266-2283.

17. Yuan X, Desiderio DM: **Proteomics analysis of prefractionated human lumbar cerebrospinal fluid**. *Proteomics* (2005) 5(2):541-550.

18. O'Farrell PH: **High resolution two-dimensional electrophoresis of proteins**. *J Biol Chem* (1975) 250(10):4007-4021.

19. Garbis S, Lubec G, Fountoulakis M: **Limitations of current proteomics technologies**. *J Chromatogr A* (2005) 1077(1):1-18. • *This paper provides a detailed overview of the challenges posed by the limitations of 2DE and MS methods.*

20. Choe LH, Dutt MJ, Relkin N, Lee KH: **Studies of potential cerebrospinal fluid molecular markers for Alzheimer's disease**. *Electrophoresis* (2002) 23(14):2247-2251.

21. Finehout EJ, Franck Z, Lee KH: **Towards two-dimensional electrophoresis mapping of the cerebrospinal fluid proteome from a single individual**. *Electrophoresis* (2004) 25(15):2564-2575.

22. Lescuyer P, Allard L, Zimmermann-Ivol CG, Burgess JA, Hughes-Frutiger S, Burkhard PR, Sanchez JC, Hochstrasser DF: **Identification of postmortem cerebrospinal fluid proteins as potential biomarkers of ischemia and neurodegeneration**. *Proteomics* (2004) 4(8):2234-2241.

23. Hsich G, Kenney K, Gibbs CJ, Lee KH, Harrington MG: **The 14-3-3 brain protein in cerebrospinal fluid as a marker for transmissible spongiform encephalopathies**. *N Engl J Med* (1996) 335(13):924-930.

24. Lee KH, Harrington MG: **Premortem diagnosis of Creutzfeldt-Jakob disease by cerebrospinal fluid analysis**. *Lancet* (1996) 348(9031):887.

25. Davidsson P, Folkesson S, Christiansson M, Lindbjer M, Dellheden B, Blennow K, Westman-Brinkmalm A: **Identification of proteins in human cerebrospinal fluid using liquid-phase isoelectric focusing as a prefractionation step followed by two-dimensional gel electrophoresis and matrix-assisted laser desorption/ionisation mass spectrometry**. *Rapid Commun Mass Spectrom* (2002) 16(22):2083-2088.

26. Puchades M, Hansson SF, Nilsson CL, Andreasen N, Blennow K, Davidsson P: **Proteomic studies of potential cerebrospinal fluid protein markers for Alzheimer's disease**. *Brain Res Mol Brain Res* (2003) 118(1-2):140-146.

27. Sihlbom C, Davidsson P, Emmett MR, Marshall AG, Nilsson CL: **Glycoproteomics of cerebrospinal fluid in neurodegenerative disease**. *Int J Mass Spectrom* (2004) 234(1-3):145-152.

28. Hakansson K, Emmett MR, Marshall AG, Davidsson P, Nilsson CL: **Structural analysis of 2D-gel-separated glycoproteins from human cerebrospinal fluid by tandem high-resolution mass spectrometry**. *J Proteome Res* (2003) 2(6):581-588.

29. Wittke S, Mischak H, Walden M, Kolch W, Radler T, Wiedemann K: **Discovery of biomarkers in human urine and cerebrospinal fluid by capillary electrophoresis coupled to mass spectrometry: Towards new diagnostic and therapeutic approaches**. *Electrophoresis* (2005) 26(7- 8):1476-1487.

30. Theodorescu D, Fliser D, Wittke S, Mischak H, Krebs R, Walden M, Ross M, Eltze E, Bettendorf O, Wulfing C, Semjonow A: **Pilot study of capillary electrophoresis coupled to mass spectrometry as a tool to define potential prostate cancer biomarkers in urine**. *Electrophoresis* (2005) 26(14):2797-2808.

31. Sassi AP, Andel F 3rd, Bitter HM, Brown MP, Chapman RG, Espiritu J, Greenquist AC, Guyon I, Horchi-Alegre M, Stults KL, Wainright A: **An automated, sheathless capillary electrophoresis-mass spectrometry platform for discovery of biomarkers in human serum**. *Electrophoresis* (2005) 26(7-8):1500-1512.

32. Chamrad D, Meyer HE: Valid data from large-scale proteomics studies. *Nat Methods* (2005) 2(9):647-648. 33. Finehout EJ, Lee KH: **An introduction to mass spectrometry applications in biological research**. *Biochem Mol Biol Education* (2004) 32(2):93-100.

34. Swanson SK, Washburn MP: **The continuing evolution of shotgun proteomics**. *Drug Discov Today* (2005) 10(10):719-725

35. Ross PL, Huang YN, Marchese JN, Williamson B, Parker K, Hattan S, Khainovski N, Pillai S, Dey S, Daniels S, Purkayastha S: **Multiplexed protein quantitation in**

Saccharomyces cerevisiae **using aminereactive isobaric tagging reagents**. *Mol Cell Proteomics* (2004) 3(12):1154-1169.

36. Gygi SP, Rist B, Gerber SA, Turecek F, Gelb MH, Aebersold R**: Quantitative analysis of complex protein mixtures using isotopecoded affinity tags**. *Nat Biotechnol* (1999) 17(10):994-999.

37. Zhang J, Goodlett DR, Peskind ER, Quinn JF, Zhou Y, Wang Q, Pan C, Yi E, Eng J, Aebersold RH, Montine TJ: **Quantitative proteomic analysis of age-related changes in human cerebrospinal fluid**. *Neurobiol Aging* (2005) 26(2):207-27

38. Zhang J, Goodlett DR, Quinn JF, Peskind E, Kaye JA, Zhou Y, Pan C, Yi E, Eng J, Wang Q, Aebersold RH, Montine TJ: **Quantitative proteomics of cerebrospinal fluid from patients with Alzheimer disease**. *J Alzheimers Dis* (2005) 7(2):125-133; discussion 173-180.

39. Carrette O, Demalte I, Scherl A, Yalkinoglu O, Corthals G, Burkhard P, Hochstrasser DF, Sanchez JC: **A panel of cerebrospinal fluid potential biomarkers for the diagnosis of Alzheimer's disease**. *Proteomics* (2003) 3(8):1486-1494.

40. Aggarwal K, Choe LH, Lee KH: **Quantitative analysis of protein expression using amine-specific isobaric tags in *Escherichia coli* cells expressing rhsA elements**. *Proteomics* (2005) 5(9):2297-2308.

41. Choe LH, Aggarwal K, Franck Z, Lee KH: **A comparison of the consistency of proteome quantitation using two-dimensional electrophoresis and shotgun isobaric tagging in *Escherichia coli* cells**. *Electrophoresis* (2005) 26(12):2437-2449.

42. Lewczuk P, Esselmann H, Groemer TW, Bibl M, Maler JM, Steinacker P, Otto M, Kornhuber J, Wiltfang J: **Amyloid β peptides in cerebrospinal fluid as profiled with surface enhanced laser desorption/ionization time-of-flight mass spectrometry: Evidence of novel biomarkers in Alzheimer's disease**. *Biol Psychiatry* (2004) 55(5):524-530.

43. Sanchez JC, Guillaume E, Lescuyer P, Allard L, Carrette O, Scherl A, Burgess J, Corthals GL, Burkhard PR, Hochstrasser DF: **Cystatin C as a potential cerebrospinal fluid marker for the diagnosis of Creutzfeldt-Jakob disease**. *Proteomics* (2004) 4(8):2229-2233.

CHAPTER 2

BIOMARKER DISCOVERY AND CLASSIFCATION OF PROTEOMICS DATA USING TWO DIFFERENT CLASSIFICATION ALGORITHMS

2.1 Introduction

A primary objective of biomarker research is to identify molecules that enable accurate clinical diagnosis of disease. One approach to identify biomarkers is through the quantification of the expression of genes or proteins that may be related to the underlying physiology of disease. The accurate quantification of these molecules is the major focus for genomics and proteomics. Enabling technologies from these fields permit detailed snapshots of gene and protein expression dynamics to be obtained. For example, genomic tools such as the microarray allow the rapid and accurate measurement of several thousands of gene transcripts in parallel. Similarly, proteomics methods, including 2D gel electrophoresis (2DGE) and mass spectrometry (MS), offer the ability to accurately measure protein expression patterns from complex protein mixtures. In recent years, biomarker research has turned to these fields with the goal of identifying and measuring the changes of genes and proteins that allow clinical diagnosis to be achieved at level of accuracy previously unobtainable.

Technologies such as microarray and 2DGE methods are often referred to as high throughput methods because of the relatively large numbers of measurements they produce. They can offer high resolution insight and, correspondingly, can also produce large amounts of data. Microarray technology can generate hundreds to an upward of millions of data points from a single hybridization. A single 2D gel of cerebrospinal fluid, for example, generates over 2,000 data points, a single measurement for every detectable protein. Similarly, an MS based shotgun proteomics experiment can generate thousands of spectra from a single sample.

Biomarker research based on high throughput methods must have the means to sift through these large data sets to discover the genes or proteins that may ultimately prove to be of high diagnostic value. To accomplish this, data mining and bioinformatics approaches capable of identifying possible relationships, and in particular, ones that may be biologically meaningful, are often applied. Here, there are a number of strategies and computational algorithms that can be implemented, and the choice of which to implement is not always straight forward.

Despite the array of algorithms in use, there are at least two common objectives. The first objective is biomarker discovery. Here, an algorithm is applied to experimental data derived from biological specimens (e.g. blood, serum, spinal fluid, etc.) that have been sampled from individuals whom have been clinically diagnosed by some gold standard, ideally one that is highly accurate. Typically, the algorithm is applied to multiple population studies where the objective is to identify genes, proteins, or some other molecule that can best discriminate these populations. A study might try to discriminate between diseased and healthy populations, or possibly even two closely related diseases. The second objective is prediction. Here, sample data from an individual is fed to an algorithm that is asked to predict its class (i.e. diseased vs. healthy). Prediction methods can be performed in conjunction with biomarker discovery or applied diagnostically. As part of the discovery process, prediction methods can be used to determine the accuracy of putative biomarkers. If class prediction is performed on a panel of known subjects, biomarker accuracy can be assessed by comparing the predicted outcomes with the known classifications. And ultimately, biomarkers that have been demonstrated to have high predictive value on large populations could be applied with a certain confidence in a clinical setting.

Additionally, there are a number of characteristics of the data set that can influence which algorithm is ultimately chosen for use in a high throughput biomarker study. The size of the data set, more specifically, the number of experimental variables (i.e. the number of genes or proteins) is one such characteristic, but other important considerations include: 1) the number of observations (e.g. number of biological replicates), 2) missing values in the data set, 3) the

number of populations being compared, and 4) the relative number of subjects contained within each of the represented populations. Characteristics such as these are important because the outcome of an analysis may be misleading or inaccurate if these factors are ignored. For example, in cases where a data set contains fewer observations than variables, statisticians often state that $n < p$. In a genomic or proteomic data set, n is the number of subjects, and p is the number of genes or proteins measured in the analysis. Dimensionality issues such as this can cause many statistical models to estimate parameters improperly and ultimately lead to overfitting [1]. Perhaps not surprisingly, p is typically very large in high throughput experiments because of the many different molecules being measured in a single experiment. Unfortunately, a study may possess only a small number of subjects, possibly because the disease of interest is rare, or simply due to low enrollment in the study. As a result, the selection of an approach that is capable of dealing with highly dimensional data is critical.

One of the other factors to consider is "missing value" which can often arise in high throughput experiments. For example, in microarray experiments, missing values occur in a data set if there is a scratch on the surface of the microarray, or dust masks the signal of several spots. In a 2D gel experiment, missing values arise if spots are unintentionally mismatched or clipped from the edge of a gel during image processing. It is thus important to select an algorithm that has the ability to deal effectively with missing values. Another consideration is the number of populations being compared which can impact algorithm selection because some algorithms can only compare two populations, while others are equipped to handle two or more. A final consideration are unbalanced data sets which arise when the study populations contain unequal numbers of observations, these may lead an analysis method to under sample from the smaller population, resulting in a loss of information. The selection of an appropriate strategy thus requires an algorithm capable of dealing with unbalanced populations through class weighting or over sampling the underrepresented population.

It is certainly plausible that a single algorithm will not possess all of the desired properties to deal with a particular data set. In such cases, the application of multiple algorithms

may be appropriate. For example, in the analysis of high dimensional data it is possible to first apply a dimension reduction method to the data set to reduce the value of p to a more manageable size, and then to apply a classification algorithm. An example of this two step approach can be found in a tumor classification study involving microarray gene expression data [2]. This study demonstrates the use of Partial Least Squares (PLS) as a dimension reduction technique followed by the use of Quadratic Discrimination Analysis (QDA) for classification. Another example of a two step dimension reduction approach is demonstrated in a recent microarray based cancer classification study where the use of two dimension reduction techniques is explored: singular value decomposition (SVD) and partial least squares (PLS), both of which are used with a penalized logistic regression (PLR) technique used for classification [1].

A two step dimension reduction is not always required, however. An alternate strategy is to implement a classification algorithm that is inherently more robust when applied to data with a high dimensional variable space. For example, many machine learning algorithms perform well with high dimensional data. Typically, machine learning algorithms fall into two categories: unsupervised and supervised. Unsupervised classification methods arrange data based on a metric of similarity and find the natural organization of data [3]. Examples of such methods are hierarchical clustering, self-organizing maps (SOM), and principal component analysis (PCA). Each implements a different metric that ultimately determines how data is organized, and inherently, unsupervised methods do not allow the number of classes to be pre-specified. Supervised classification methods, on the other hand, allow the number of classes to be pre-specified. In other words, the algorithms have *a priori* knowledge of the number of groups that exist in a data set, and attempt to classify data accordingly. Some examples of supervised methods are support vector machines (SVM), artificial neural networks (ANN), decision tree learning, and random forests (RF), among others.

While unsupervised methods are useful for the identification of the underlying organization of the data, supervised learning methods are often most aligned with the goals of

biomarker studies for several reasons. First, the number of classes (e.g. diseased vs. healthy) is typically known in advance. Second, supervised learning methods have the ability to discover new biomarkers, as well as to predict the class of unknown samples.

In this chapter, two supervised classification methods are implemented. The first is Fisher's Linear Discrimination Analysis (LDA), and the other is Random Forest (RF). Additionally, both methods are implemented using a gene shaving (GS) approach. GS is a process by which training models, such as those constructed by LDA and RF, are iteratively fit to training data while removing the least significant variables from the analysis (i.e. shaved from the training set) after each iteration. Here the methods were based on LDA and RF because of their demonstrated ability to classify data with a large number of variables. A gene shaving approach was chosen because this approach simplifies variable selection by quickly identifying the most important variable combinations and it has also been used in other biomarker studies with good success.

2.1.1 Fisher's Linear Discrimination Analysis

LDA was first proposed by Sir Ronald Fisher in 1936 [4]. This model closely resembles PCA, its unsupervised counterpart, but differs in that it assigns a class label to variables and attempts to find a linear combination of variables that best separate represented classes. While LDA is considered an excellent dimension reduction method, it still may perform poorly when $n << p$, and may also be sensitive unbalanced data sets [5].

2.1.2 Random Forests

The RF algorithm was conceived more recently than LDA and made publicly available to research communities by Leo Breiman and Adele Cutler. It has been applied to a variety of applications including the differential expression analysis of microarray data. As described by Breiman and Cutler, class prediction is achieved by the RF algorithm through the construction of

many classification trees. Each classification tree comprises a part of a 'forest' and is 'grown' from training data. As part of RF training, many classification trees are grown. To predict the class of an unknown sample, the sample data is run down each of the trees within the forest resulting in a single classification, or "vote", from each tree. The overall classification is determined by selection of the class that receives the most "votes". The algorithm possesses a number of characteristics that make it a robust and ideal method for dealing with high throughput biomarker data. First, it performs an efficient and fast analysis of very large data sets and has the ability to analyze a very large number of variables. Unlike LDA, RF performs well even when $n < p$. Second, RF effectively manages missing data and unbalanced data sets. And lastly, in addition to class prediction, it estimates the importance of variables used in classification [6].

Like most supervised classification models, both LDA and RF implement learning algorithms that are first trained to recognize classes present in the data set, and then are asked to predict the class of unknown samples. In practice, trained models must also be validated on a test-set that contains samples with known classes; the subject classes are not disclosed to the classifier and as a result, the model can be evaluated for accuracy. Importantly, in the context of biomarker discovery, validated models mean it is possible to extract the most important variables used in classifying sample data.

In the following analyses, LDA and RF are applied to two paired-class data sets that compare Alzheimer's disease (AD) subjects with healthy subjects (N), and AD subjects to Parkinson's disease (PD) subjects. From these analyses, several putative biomarkers are identified, and the accuracy of these putative biomarkers is assessed.

2.2 Methods

2.2.1 Protein Quantification and Identification

Protein expression data used in these analyses were generated from 2D gel electrophoresis (2DGE) separations of cerebrospinal fluid (CSF) samples. Detailed experimental procedures can be found in Finehout *et al* [7]. Briefly, each CSF sample was prepared and then separated using

2DGE; resulting gels were then stained to resolve the separated protein spots. Stained gels were scanned using a laser fluorescence imaging scanner, generating a TIFF image for each gel. These images were then analyzed using ImageMaster 2D Platinum (GE Healthcare Life Sciences) to quantify the protein expression of gel spots. The determined protein expression values were expressed as a percent volume (% vol.) of the total spot volume for each gel as a normalization step. Protein spots of interest were excised from the gels and identified using MS.

2.2.2 Threshold Method

The nature of the experimental procedures is such that with image analysis and spot detection procedures, partially automated, they and are not perfect. Very low intensity spots are often identified during image analysis and may be spots that are poorly represented among all gels analyzed in a given experiment. Such spots are particularly problematic because of background noise and may unnecessarily contribute to variability within the data set. To minimize the possibility of background effects due to the presence of very low intensity spots, a threshold was applied to remove these spots from data sets included in these analyses. For n gels and i spots, $i_1 \ldots i_n$ spots are excluded if T > mean ($i_1 \ldots i_n$), where T is a specified threshold value. T is expressed as a spot % volume, and for all analyses described, T = 0.001.

2.2.3 GSFLD and RFshave Implementation

The GSFLD algorithm implements a gene shaving approach using LDA. The algorithm was first described by Jiang *et al*. where it was used to identify possible lung adenocarcinoma biomarkers from microarray data [8]. When applied to the data set which first fits all variables to determine which are the most important, and using an importance value calculated for each variable, the 10% least important variables are iteratively removed from the model. While LDA is capable of classifying multiple classes, GSFLD (version 2.0) can only be applied to two-class analyses. GSFLD requires only a single training-set as input and classification error-rates are determined

by a leave-one-out cross-validation strategy. As the name implies, the model repeatedly samples all but one observation from the training-set; the remaining observation is used to test the classification accuracy of selected variables. GSFLD runs as a command-line Microsoft Windows executable file and is available upon request by its authors [8].

The RF algorithm applied in this section (RFshave) is written in the R statistical programming language [9] and was developed specifically for this investigation. It implements the R package, randomForest (version 4.5-16) written by Andy Liaw and Matthew Wiener, and is based on the FORTRAN release of the algorithm (version 5.0) by Breiman and Cutler. The R and FORTRAN versions are essentially identical; however the R version has a limited ability to deal with unbalanced data sets. Like GSLDA, RFshave implements an iterative variable shaving strategy. But differently, it removes it removes the single least important variable after each iteration, rather than the 10% least important variables.

2.2.4 Leave-One-Out Cross-Validation

To determine the prediction accuracy of variables selected by GSLDA and rfShave, sensitivity and specificity values were calculated using a leave-one-out (LOO) strategy. This strategy was implemented in R using LDA (package: MASS) to assess prediction accuracy. Training and test data are supplied to the LDA model iteratively, such that in each cycle (where the number of cycles equals n subjects), one subject is removed from the training-set and classified by the model. Each subject is classified only once and otherwise remains in the training-set. The outcomes of the LDA predictions is used to calculate the number of True Positives, True Negatives, True Positives, and False Positives, represented by TP, TN, TP, and FP, respectively. Sensitivity is calculated as: TP /(TP + FN), and specificity as: TN/(FP + TN).

2.3 Results

Two sets of analyses were performed using 2DGE protein expression data from CSF sampled from AD, PD, and N subjects. In the first set, biomarker discovery was performed by applying GSFLD and RFshave to a paired data set containing AD and N subjects. This data set is referred to the AD x N data set and it contained 2DGE protein expression measures from a total of 19 subjects (10 AD and 9 N). These analyses were then repeated, but a paired data set containing AD and PD subjects was analyzed instead. This data set, AD x PD, contained expression measures from 19 subjects (9 AD and 10 PD). A threshold was applied to both the AD x N and AD x PD data sets (T=0.001).

2.3.1 GSLDA and RFshave: Biomarker Discovery using AD and N subjects

The threshold applied to the raw AD x N data set excluded 100 variables, resulting in a new data set containing expression measures for 1738 variables. Both GSLDA and RFshave were applied to this new data set and both analyses resulted in a number of significant variables to be identified (Table 2.1). The predication accuracy of the variables was assessed by selecting and validating two variable sets (p=4 and p=8) from each analysis (i.e. two sets from the GSLDA results and two sets from the RFshave results). The results of this validation are shown in the 2x2 tables in Figure 2.1. For all variable sets selected, the classification accuracy of AD and N subjects was achieved with a sensitivity and specificity of 100% and 100%, respectively.

The proteins and corresponding spot IDs represented by the variables in Table 2.1 are specified in Table 2.2. For the GSLDA results, five of the spots had been previously identified as transthyretin, vitamin D binding protein, alpha-1 antitrypsin (two spots), and alpha-1 B glycoprotein. Four of the spots were unknown (Variables: V1356, V714, V942, and V312). For the RFshave results, transthyretin, vitamin D binding protein, alpha-1 antitrypsin (two spots), and alpha-1 B glycoprotein were also identified along with four unknowns (Variables: V1356, V42, V942, and V312).

Table 2.1 Significant variables identified by GSFLA and RFshave: AD x N.

Shown in these tables are the top *p* most significant variables identified by GSFLD (A) and RFshave (B). As part of the gene-shaving process, *p* variables were used for classification of the AD and N classes after the least significant variables were iteratively removed from the training-set. The errors shown in **A** were calculated internally by GSFLD using a leave-one-out cross-validation (GSFLD-LOO-CV) and the out-of-bag (OOB) errors shown in **B** were calculated internally by the RF algorithm. Both the GSFLD-LOO-CV and OOB errors are an indicator of classification accuracy.

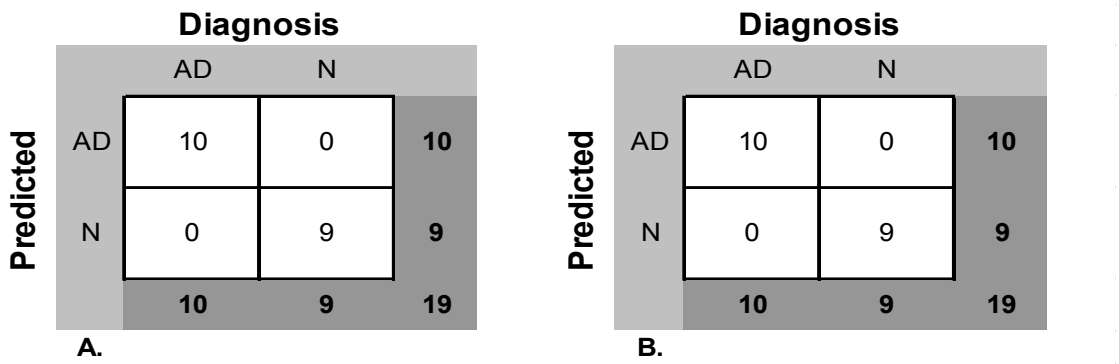
p	Error	Varaiables
2	2	V719 V1356
3	2	V719 V1356 V700
4	2	V719 V1356 V700 V714
5	2	V719 V1356 V700 V714 V967
6	2	V719 V1356 V700 V714 V967 V998
7	2	V719 V1356 V700 V714 V967 V998 V194
8	2	V719 V1356 V700 V714 V967 V998 V194 V942
9	2	V719 V1356 V700 V714 V967 V998 V194 V942 V312

A.

p	OOB Error %	Varaiables
2	0	V719,V1356
3	5.263	V719,V1356,V967
4	0	V719,V1356,V312,V967
5	5.263	V719,V1356,V967,V312,V700
6	5.263	V719,V1356,V700,V967,V312,V194
7	5.263	V719,V1356,V967,V312,V700,V194,V998
8	0	V719,V1356,V967,V312,V700,V194,V998,V42
9	5.263	V719,V1356,V700,V312,V967,V194,V998,V42,V942

B.

GSFLD:



RFshave:

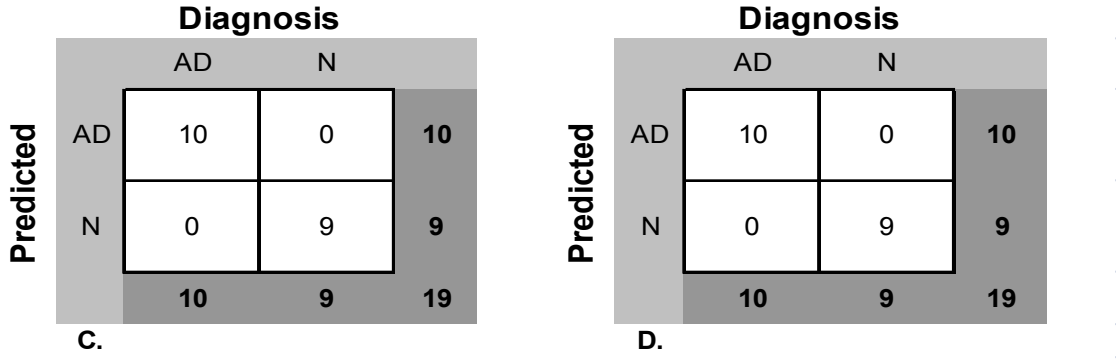


Figure 2.1 Leave-one-out cross validation results: AD vs. N

These 2x2 tables report the accuracy of class predictions as evaluated by the leave-one-out cross-validation. The validation was performed using sets of discriminators identified as significant by the GSFLD (A, B), and RFshave (C, D) analyses. The variables selected for use in each validation correspond to $p = 4$ and 8 (Table 2.1): A) V719, V1356, V700, and V714, B) V719, V1356, V700, V714, V967, V998, V194, and V942, C) V719, V1356, V967, and V312 and D) V719, V1356, V967, V312, V700, V194, V998, and V42. C) and D). A total of 19 subjects were included in each validation, 10 AD and 9 N. The values in each white box represent predictive rates, clockwise from the upper left: TP, FP, TN, and FN.

Table 2.2 Variable Identities: AD x N.

Shown are the 2DGE spot IDs and protein identities corresponding to variables found to be significant by GSFLD (A), and RFshave (B).

Variable	Spot ID	Protein Identity
V719	725	Transthyretin
V1356	1378	Unknown
V700	706	Vitamin D Binding Protein
V714	720	Unknown
V967	974	alpha-1 antitrypsin
V998	1005	alpha-1 antitrypsin
V194	196	alpha-1 B glycoprotein
V942	949	Unknown
V312	314	Unknown

A.

Variable	Spot ID	Protein Identity
V719	725	Transthyretin
V1356	1378	Unknown
V700	706	Vitamin D Binding Protein
V312	314	Unknown
V967	974	alpha-1-antitrypsin
V194	196	alpha-1-B glycoprotein
V998	1005	alpha-1-antitrypsin
V42	42	Unknown
V942	949	Unknown

B.

2.3.2 GSLDA and RFshave: Classification of AD and PD subjects

The threshold applied to the raw AD x PD data set excluded 174 variables, resulting in a new data set containing expression measures for 1765 variables. Both GSLDA and RFshave were applied to this new data set and both analyses resulted in a number of significant variables to be identified (Table 2.3). As with the AD x N analyses, the predication accuracy of these variables was assessed by selecting and validating two variable sets (p=4 and p=8) from each analysis. For the variable set resulting from the GSLDA analysis, where p=4, the sensitivity and specificity were determined to be 88.89% and 100%, respectively. For the variable set where p=8, the classification accuracy of AD and PD subjects was achieved with a sensitivity and specificity of 77.78% and 100%, respectively. For both variable sets selected from the RFshave results, the classification accuracy of AD and PD subjects was achieved with a sensitivity and specificity of 100% and 100%, respectively. The 2 x 2 tables for these validations are shown in the in Figure 2.2.

The proteins and corresponding spot IDs represented by the variables shown in Table 2.3 are specified in Table 2.4. For the GSLDA results, four of the spots had been previously identified as Complement C3, alpha-1 antitrypsin, albumin and transferrin. Five of the spots were unknown (Variables: V94, V1285, V1039, V665, V747).

For the RFshave results, only 3 spots were previously identified by MS, these were identified as: albumin, alpha-1 antitrypsin, and prostaglandin D2 synthase, along with five unknowns (Variables: V1039, V94, V1285, V132, and V747).

Table 2.3 Significant variables identified by GSFLD and RFshave: AD x PD

Shown in these tables are the top *p* most significant variables identified by GSFLD (A) and RFshave (B) and corresponding classification error. Error rate is measure of classification accuracy and determined by GSFLD-LOO-CV and OOB for GSFLD and RFshave, respectively.

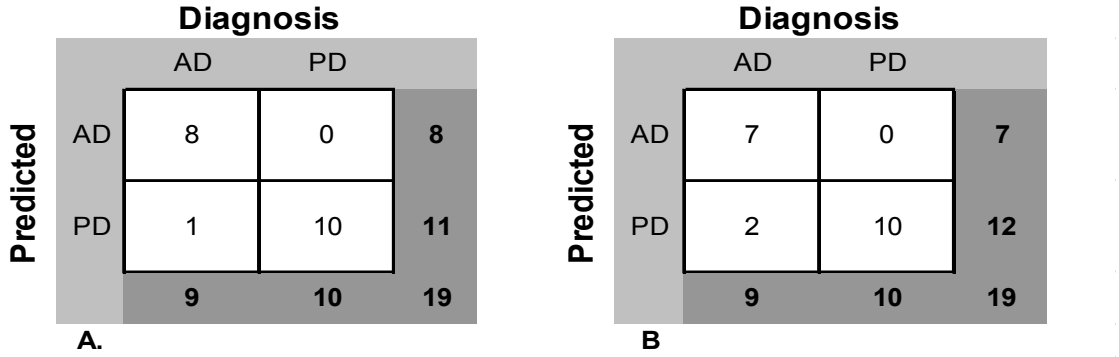
p	Error	Varaiables
2	3	V94 V244
3	3	V94 V244 V1285
4	2	V94 V244 V1285 V967
5	2	V94 V244 V1285 V967 V285
6	1	V94 V244 V1285 V967 V285 V1039
7	1	V94 V244 V1285 V967 V285 V1039 V665
8	1	V94 V244 V1285 V967 V285 V1039 V665 V747
9	1	V94 V244 V1285 V967 V285 V1039 V665 V747 V54

A.

p	OOB Error %	Varaiables
2	10.53	V1039,V285
3	5.263	V285,V1039,V94
4	5.263	V285,V1039,V94,V1285
5	5.263	V285,V1039,V94,V967,V1285
6	5.263	V285,V1039,V94,V967,V876,V1285
7	5.263	V285,V1039,V94,V876,V967,V1036,V1285
8	5.263	V285,V1039,V94,V967,V1036,V876,V1285,V747
9	5.263	V285,V1039,V94,V967,V1285,V876,V1036,V132,V747

B.

GSFLD:



RFshave:

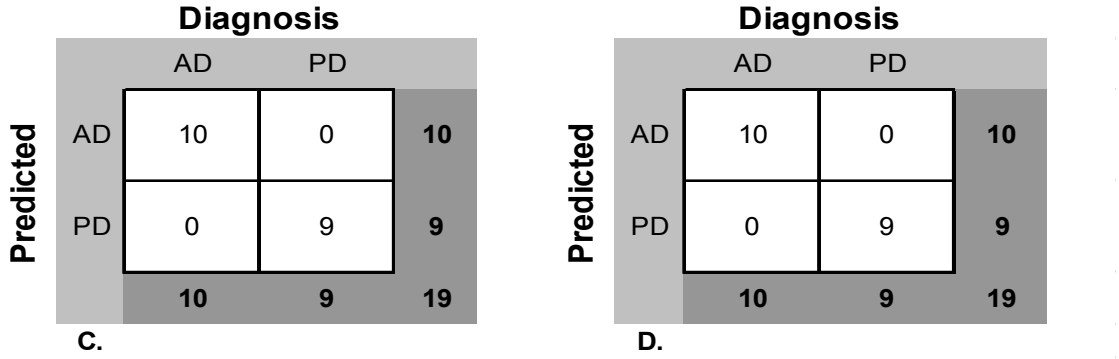


Figure 2.2 Leave-one-out cross validation results: AD vs. PD

These 2x2 tables report the accuracy of class predictions as evaluated by the leave-one-out cross-validation. The validation was performed using sets of discriminators identified as significant by the GSFLD (A, B), and RFshave (C, D) analyses. The variables selected for use in each validation correspond to $p = 4$ and 8 (Table 2.3): A) V94, V244, V1285, and V967, and B) V94, V244, V1285, V967, V285, V1039, V665, and V747, C) V285, V1039, V94, and V967, and D) V285, V1039, V94, V967, V1036, V876, V1285, and V747A total of 19 subjects were included in validation, 9 AD and 10 PD.

Table 2.4 Variable Identities: AD x PD

Shown are the 2DGE spot IDs and protein identities corresponding to variables found to be significant by GSFLD (A), and RFshave (B).

Variable	Spot ID	Protein Identity
V94	94	Unknown
V244	246	Complement C3
V1285	1304	Unknown
V967	974	alpha-1 antitrypsin
V285	287	Albumin
V1039	1046	Unknown
V665	670	Unknown
V747	753	Unknown
V54	54	Transferrin

A.

Variable	Spot ID	Protein Identity
V285	287	Albumin
V1039	1046	Unknown
V94	94	Unknown
V967	974	alpha-1-antitrypsin
V1285	1304	Unknown
V876	883	Prostaglandin D2 Synthase
V1036	1043	Albumin
V132	132	Unknown
V747	753	Unknown

B.

The validation results of the four experiments presented in the previous sections are summarized in Table 2.5. A comparison of the variables and corresponding proteins identified by both GSFLD and RFshave were also made to assess outcome similarity. A number of similar variables from both the AD x N and AD x PD data sets were identified by both methods. These similarities are summarized in Table 2.6.

Table 2.5 Sensitivity and Specificity outcomes from each analysis.

Method	Comparison	p	Senstivity	Specificity
GSFLD	AD x N	4	100%	100%
GSFLD	AD x N	8	100%	100%
GSFLD	AD x PD	4	89%	100%
GSFLD	AD x PD	8	78%	100%
RFshave	AD x N	4	100%	100%
RFshave	AD x N	8	100%	100%
RFshave	AD x PD	4	100%	100%
RFshave	AD x PD	8	100%	100%

Table 2.6 Variables identified by both GSLDA and RFshave

Shown are the significant variables (and corresponding Spot IDs and protein identities) identified by both GSLDA and RFshave. Table A represents variables from the AD x N data set; Table B, those found from the AD x PD data set. Variables shown were among the top 9 most significant variables identified by each method.

A

Variable	Spot ID	Protein Identity
V194	196	alpha-1-B glycoprotein
V700	706	Vitamin D Binding Protein
V719	725	Transthyretin
V942	949	Unknown
V967	974	alpha-1-antitrypsin
V998	1005	alpha-1-antitrypsin
V1356	1378	Unknown

B

Variable	Spot ID	Protein Identity
V94	94	Unknown
V285	287	Albumin
V747	753	Unknown
V967	974	alpha-1 antitrypsin
V1039	1046	Unknown
V1285	1304	Unknown

It was of interest to determine whether any of the proteins identified in these analyses have been previously identified as possible AD or PD biomarkers. These results are summarized in Table 2.7.

Table 2.7 Proteins previously identified as putative AD or PD biomarkers

Protein Identity	Comparison	Reference
Albumin	AD x PD	AD [10]
alpha-1 antitrypsin	AD x N, AD x PD	AD [10]
alpha-1 B glycoprotein	AD x N	AD [10]
Complement C3	AD x PD	
Prostaglandin D2 Synthase	AD x PD	AD [10]
Transferrin	AD x PD	AD [10]
Transthyretin	AD x N	AD [10]
Vitamin D Binding Protein	AD x N	AD, PD [11]

2.4 Discussion and Conclusion

Both methods applied in this study identified several proteins that have been previously identified as potential biomarkers (Table 2.7). Additionally, the predictive value of these biomarkers identified by both methods was quite high as evaluated by the LOO cross-validation. The proteins identified by the RFshave approach, had a slightly higher predictive value than that of the GSLDA method when classifying the subjects in the AD x PD data set (Table 2.5). While this may indicate that the RFshave approach identified a superior set of classification variables, it is interesting to note that the variables identified by the RFshave and the GSLDA methods are actually quite similar (Table 2.6). In fact, when considering only the top 9 variables identified by both the RFshave and GSLDA methods, 78% and 66% of the variables were identical for the AD x N and AD x PD data sets, respectively. The consistency of the outcomes between these two different methods helps support the validity of these results.

One limitation of the study presented here is that both data sets possess only 19 subjects. This limitation arose from an effort to maintain balanced data sets, and in this study, the numbers of N and PD subjects were scarce. Data sets possessing a small number of observations,

regardless of the number of variables it possesses, may not fully represent the population it is intended to describe, and it is possible that the predictive value of the proteins identified in study would be less when applied to larger populations.

In conclusion, this work has demonstrated the use of two valuable biomarker discovery and prediction methods. Both methods identified several proteins that were able to predict disease state with high sensitivity and specificity. Additionally, this work adds further support to the potential importance of several putative biomarkers that may aid in the prediction and clinical diagnosis of Alzheimer's and Parkinson's disease.

REFERENCES

1. Shen L, Tan E: **Dimension reduction-based penalized logistic regression for cancer classification using microarray data.** *IEEE/ACM Trans Comput Biol Bioinform* 2005, 2(2):166-175.

2. Nguyen D, Rocke D: **Tumor classification by partial least squares using microarray gene expression data.** *Bioinformatics* 2002, 18(1):39-50.

3. Phan J, Quo C, Wang M: **Functional genomics and proteomics in the clinical neurosciences: data mining and bioinformatics.** *Prog Brain Res* 2006, 158:83-108.

4. R.A. F: **The Use of Multiple Measurements in Taxonomic Problems.** *Annals of Eugenics* 1936, 7:179-188.

5. W.N. V: **Modern Applied Statistics with S.**: Springer; 2002.

6. Breiman, L. **Random Forests.** In *Machine Learning*; Kluwer Academic Publishers; 2001; Vol. 45; pp 5-32.

7. Finehout EJ, Franck Z, Lee KH: **Towards two-dimensional electrophoresis mapping of the cerebrospinal fluid proteome from a single individual**. *Electrophoresis* (2004) 25(15):2564-2575.

8. Jiang H, Deng Y, Chen H, Tao L, Sha Q, Chen J, Tsai C, Zhang S: **Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes.** *BMC Bioinformatics* 2004, 5:81.

9. R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/

10. Puchades M, Hansson S, Nilsson C, Andreasen N, Blennow K, Davidsson P: **Proteomic studies of potential cerebrospinal fluid protein markers for Alzheimer's disease.** *Brain Res Mol Brain Res* 2003, 118(1-2):140-146.

11. Abdi F, Quinn J, Jankovic J, McIntosh M, Leverenz J, Peskind E, Nixon R, Nutt J, Chung K, Zabetian C *et al*: **Detection of biomarkers with a multiplex quantitative proteomic platform in cerebrospinal fluid of patients with neurodegenerative disorders.** *J Alzheimers Dis* 2006, 9(3):293-348.

CHAPTER 3*

ITRAQPAK: AN R BASED ANALYSIS AND VISUALIZATION PACKAGE FOR 8-PLEX ISOBARIC PROTEIN EXPRESSION DATA

3.1 Introduction

The rapidly expanding field of proteomics is advancing the ways in which protein expression dynamics can be studied. Contributing to these advances is the proliferation of mass spectrometry (MS) based shotgun proteomics methods that have been introduced in recent years. Shotgun methods allow the rapid profiling of complex protein mixtures by coupling high resolution separation methods, such as HPLC, with the accurate quantitation and identification capacity of MS based technologies. There are a number of methods to quantify protein expression from shotgun experiments including those based on isotope-coded affinity tag and isobaric tag for relative and absolute quantitation (iTRAQ™) technologies [1].

One of the more recently introduced methods is iTRAQ (trademarked by Applied Biosystems), which permits multiplex quantitation of up to eight complex protein samples in a single analysis. The experimental workflow for isobaric tagging based quantitation is similar to traditional single sample HPLC-MS based quantitation methods [2], however multiplexing is achieved through the use of isobaric labeling reagents that allow multiple samples to be pooled and quantitated independently. The first version of the technology permitted four samples to be multiplexed and the newer versions permit eight sample multiplexing [3]. The 4-plex reagents consist of four reporter ions (isobaric tags) which are designated as: 114, 115, 116, and 117, and the 8-plex reagents consist of these four, plus four additional tags, designated as: 113, 118, 119 and 121. Briefly, unlabeled protein samples are trypsin-digested, labeled using isobaric tags, then separated by liquid chromatography (perhaps multiple dimensions), and finally, peptides are

* Chapter 3 was published in *Briefings in Functional Genomics and Proteomics* (2008) 7(2):127-35.

quantified and sequenced by tandem MS (MS/MS). The isobaric tags covalently bind to the N-terminus and lysine (Lys) side chain of peptides during labeling and enable multiplexing because they each have the same charge and overall mass, but produce different low mass signatures upon MS/MS [4, 5]. This unique characteristic allows otherwise identical peptides from different samples to be detected as a single peak by MS and produce a single set of sequencing ions in MS/MS, while maintaining the quantitation information from the different samples. Absolute and relative quantitation is achieved by determining the MS/MS spectra peak areas associated with each of the reporter ions and comparing them.

As shotgun methods continue to be applied to scientific research studies, it is also important to develop tools capable of analyzing the data they generate. In this report, we describe the features, and demonstrate the application, of one such a tool which is intended for use with 8-plex expression data. The tool is developed as a package for use in the R statistical programming environment, and is called: iTRAQPak. It performs routine data transformation tasks associated with isobaric tag-based shotgun proteomics expression data analysis, and also implements more complex analytical, statistical, and visualization functionality that may allow important biological relationships to be identified. We apply this tool to the analysis of 8-plex expression data collected in association with a longitudinal study of two Alzheimer's disease (AD) patients undergoing a passive immunization treatment as part of a Phase I drug trial. The results of this analysis are used to demonstrate the utility of several iTRAQPak functions and highlight its visualization approach which provides a novel view of 8-plex expression data.

3.2 Implementation and Overview

The iTRAQPak package was developed in the R programming language [6]. R is a rich statistical, data analysis, and visualization environment that is widely used and freely available under the GNU General Public License. iTRAQPak has been developed and tested under the Windows computing environment (Microsoft Windows Server 2003), however R is available

under a variety of operating system environments. Package functions are run from the R command line.

Expression data are imported into R using the iTRAQPak function *LoadData*. The function expects data to contain a number of required variables; these are shown in Table 3.1. Expression data with corresponding variable columns can be generated through the use of software such as GPS Explorer (Applied Biosystems) with Mascot (Matrix Science) integration. While the set of required variables is fixed, the columnar order of the input variables is customizable through command line parameter options which are accessible during data import.

The package provides a number of data transformation options which can be selectively applied to imported data sets using user supplied parameter options. Transformation options include: isotope impurity correction, sample normalization, peak scaling, and log transformation. Some form of impurity correction is recommended by the manufacturer for iTRAQ data analysis because the labeling reagents contain trace levels of isotopic impurities that cause variations in the MS peak intensities. Applying the impurity correction transformation to a data set adjusts peak intensities as specified in the 'Certificate of Analysis' by the reagent manufacturer. Table 3.2 defines the default peak area correction factor applied to each peak type; however these values are customizable by modifying parameter input supplied to the package. Various normalization procedures can be applied to sample data, and these are intended to correct between sample variation that may arise due to

Table 3.1 Required columns and corresponding R data types

Column	Description	R Data Type
1	Plate Number	numeric
2	Spot Label	character
3	Protein Name	character
4	Accession Number	character
5	Modification	character
6	Ion Score C.I. %	numeric
7	Best Peptide Sequence	character
8	Start Sequence Position	numeric
9	End Sequence Position	numeric
10	Ion Score	numeric
11	Calculated Mass	numeric
12	Observed Mass	numeric
13	Match Error PPM	numeric
14	Area 113	numeric
15	Area 114	numeric
16	Area 115	numeric
17	Area 116	numeric
18	Area 117	numeric
19	Area 118	numeric
20	Area 119	numeric
21	Area 121	numeric

Table 3.2 Default correction factors

TAG	-2	-1	+1	+2
113	0	2.5	3	0.1
114	0	1	5.9	0.1
115	0	2	5.6	0.1
116	0	3	4.5	0.1
117	0.1	4	3.5	0.1
118	0.1	2	3	0.1
119	0.1	2	4	0.1
121	0.1	2	3	0.1

experimental procedure, rather than from biological differences. Normalization methods that can be applied are based on: 1) sample mean, 2) sample median, and 3) Lowes regression [7]. Applying peak scaling expresses peak areas as a relative quantitation rather than absolute, and parameter settings allow up to two of the tags to be specified as a baseline for relative comparison. For example, with the default peak scaling parameters, areas associated with tags 115, 117, and 119 are expressed relative to tag 113 associated areas, and the remaining areas are expressed relative to 114 associated areas. Finally, scaled data can be Log2 transformed in preparation for statistical analysis or to represent expression increases and decreases as positive and negative values to allow for a more intuitive visual interpretation when viewed graphically.

When several peptides with identical sequences are identified during quantitative shotgun data analysis, one strategy is to expresses their peak areas as a single averaged area. The advantage of averaging peak areas is that expression values are derived from more than one observation and, additionally, data set complexity is also reduced, thus allowing for a more simplified interpretation. However, condensing data using such an approach over-simplifies the complexity related to posttranscriptional and posttranslational gene expression [3]; and as a consequence, important biological information can be lost. While peptides may be indistinguishable at the amino acid (AA) sequence level, they may be distinguished when posttranslational modifications (PTMs) or other modifications are considered. Ignoring this difference may result in important biological relationships being overlooked. Further, two peptides, identical at the sequence level, may actually be from two highly homologous genes that have different biological roles. Averaging, in this case, makes little sense from a biological perspective.

With this in mind, iTRAQPak implements a multitiered strategy to avoid such loss of information by representing data at an uncondensed level plus three condensed levels: 1) the peptide level, 2) the peptide-modifications level, and 3) the protein level. Data are condensed by averaging grouped peak areas defined by these levels. At the peptide level, peak areas are distinguished by amino acid sequence, and similarly, at the peptide-modifications level, peptides

are distinguished by sequence and modification. At the protein level, peak areas are distinguished by accession number (GI). Finally, at the uncondensed level, as the name implies, no averaging is performed.

Peptide expression maps are generated using the iTRAQPak *PlotExpression* function. The expression maps present several different views of expression data, where the data are condensed to different levels within several of the views. Uncondensed expression data are displayed using two different view styles. The first displays expression data using heat maps and the second, using expression plots. Heat-maps are widely used to represent expression change, for example, red/green heat-maps are commonly used to represent microarray gene expression data, where red typically represents an increase in expression and green, a decrease [8]. A similar red/green strategy is used to visually represent relative log-fold changes in peptide expression within iTRAQPak peptide expression maps. Expression plots are displayed as an alternative to the heat-map representation, but rather than using color to represent expression change, two dimensional line plots are used, where the y-axis represents relative log-fold changes in peptide expression. For both formats, the uncondensed view displays expression values of every peptide detected. Expression plots are also used to represent condensed views for both peptide and protein level expression data and are a standard display within expression maps. Because there are a number of modifications that may be of interest in a research study, condensed peptide-modifications level data can be optionally displayed within the heat-map view, using user supplied parameters.

Expression map functions also allow peptide-modifications level expression data to be presented in an uncondensed form. Here, the uncondensed peptides displayed in the heat-map view are simply highlighted. Specifying modifications of interest is achieved by supplying plotting functions with a parameter list of modification-codes. Parameter functions also allow peptides that lack a particular modification to be highlighted. As an example, it is possible to highlight peptides that possess a Lys residue but lack a [K]-iTRAQ tag. In the current implementation (1.0.7), it is possible to highlight peptides with these modifications: Oxidation,

MMTS, [K]-iTRAQ, and [N-Term]-iTRAQ. Additionally, because the plotting functions search the Modifications column of the input data set, modifications that can be specified are limited to those present in this column.

The use of heat-maps in the peptide expression maps is a method implemented to reduce the visual complexity of the underlying expression data. Another method that is implemented is self organizing map (SOM) clustering, which is widely used as a data visualization method for mapping high-dimensional non-linear data into a lower dimensional visualization space. Applied to expression plots, it allows complex expression patterns to be color coded based on similarity. The package implements SOM clustering to color code peptide expression patterns using the SOM package available in R.

Several data quality functions are implemented in the iTRAQPak package, two of which are: *TagSummary* and *LabelingEfficiency*. When applied to the raw expression data, *TagSummary* applies several transformation steps to the expression data, groups it by isobaric tag, and plots the transformed data to PDF. The PDF output consists of five plots, the first shows Log2 raw data (i.e. the data are not normalized, scaled, nor corrected, but are Log2 transformed) peptide expression values grouped by isobaric tag, and the second shows the effects of scaling, and the remaining three show the effects of mean, median and Lowes regression normalization. The *LabelingEfficiency* function uses peptide sequence and modification information to evaluate the efficiency of labeling reactions and then plots the results. It is assumed that under ideal conditions, the labeling reactions will go to completion such that all N-terminal and Lys residues are labeled with an iTRAQ tag. While this may not be the case under normal experimental conditions, it is possible to quantify the abundance of partially and fully labeled peptides. A partially labeled peptide is defined as one that has been labeled with at least one isobaric tag, but contains other possible labeling sites that have not been labeled. As an example, a peptide may contain a Lys residue in addition to its N-terminal residue; if it were partially labeled, then only the Lys group would be labeled. Using peptide sequence information, it is possible to assess which peptides contain Lys, in addition to N-terminal, residues, and using high confidence

search results, it is possible to assess which peptide residues have been labeled by an isobaric tag. Using this information, a relative calculation can be made by summing the areas of all partially labeled peptides and expressing this value as fraction of the summed area of fully labeled peptides.

3.3 Methods

To assess the utility of the iTRAQPak package, cerebrospinal fluid samples from two probable AD subjects were analyzed using 8-plex technology and methods as described previously [3]. Both of these subjects were enrolled in a Phase I drug trial investigating the effects of intravenous immunoglobulin as an AD treatment. As part of a longitudinal study, both patients received ongoing treatment with IVIg and cerebrospinal fluid (CSF) samples were collected (with appropriate consent) by lumbar puncture from each patient over the course of the study.

In this study, four CSF samples from each AD patient were collected for protein expression analysis. These samples were collected at four different time points: at a baseline timepoint at the beginning of the drug treatment regimen, after three months and six months of treatment, and then after a subsequent three month drug washout period (i.e. after 0, 3, 6, and 9 months). CSF was collected and analyzed as described previously [3].

In the R programming environment, the iTRAQPak package was imported and several package functions applied to the isobaric tag-based protein and peptide expression data. As the first step of the analysis, the expression data set was imported into R using iTRAQPak's *LoadData* function. Next, using the *TransformData* function, peptide expression data were corrected for isotopic impurities, median normalized, and scaled. Using scaling parameter options, peptide expression values were scaled relative to the "0" time point (T1) for both patients. Also using the *TransformData* function, scaled values were then expressed as Log2 values and subsequently used for statistical and visualization analyses. Statistical analyses applied in this study consisted of three-way analysis of variance (ANOVA) and was performed

in R using the *lm* and *anova* functions. Peptide expression maps were generated using the *PlotExpression* function. To highlight peptides containing modifications of interest, the *pepMod* function was applied to expression data and expression maps were regenerated. Labeling Efficiency plots were created using the *LabelingEfficiency* function. Details on how to invoke package functions and parameter details are supplied within the iTRAQPak help files.

3.4 Results and Discussion

The 8-plex iTRAQ data set contained expression values for 1187 peptides (GPS ion score confidence interval 95% or greater) derived from 167 proteins (false positive rate of 0.8%). A complete list of proteins and peptides is provided in [3]. Here, we report on the development of the iTRAQPak software package and its application to this data set.

Expression data were imported into R, and several transformation steps were applied to the imported data as described in the methods section. Mean peptide expression levels for each of the labeled samples were inspected before and after normalization (Figure 3.1). Before normalization, the mean expression values for peptides labeled with the 113 tag were noticeably lower than the mean expression values for the peptides labeled by the other 7 tags. After normalization, this difference was less noticeable. However, when inspecting the scaled values, it was observed that the expression values for samples scaled to the 113 tag labeled sample were more variable than the expression values scaled to the 114 tag labeled sample. This was also observed for mean and Lowes regression normalization methods (data not shown). While normalization corrected for some of the between sample variability, it did not correct for all, suggesting a more complex level of variability exists for this data.

Using iTRAQPak functions, peptide expression maps were created for each unique protein identified in the normalized and transformed data set. As an example, the map for albumin is shown in Figure 3.2. Maps were output from R as high-resolution PDF files which allowed them to be easily viewed using Acrobat Reader (Adobe Systems). Acrobat Reader's search functions were useful to quickly search for GI numbers of interest, and additionally, the zoom and scroll

features allowed the features of the plots to be more closely inspected. Detailed views of heat-map and expression plots for albumin are shown in Figures 3.3 and 3.4, respectively.

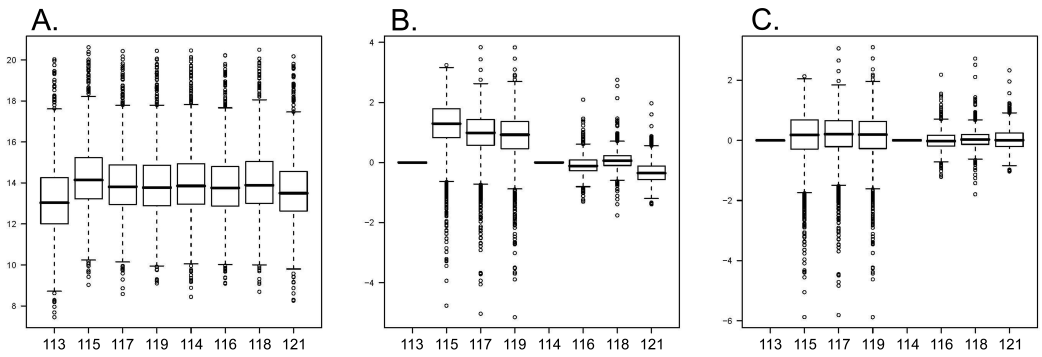


Figure 3.1 Normalization, Correction, and Scaling of data

All data presented are in Log base 2. Panel A shows raw peak area data from each isobaric tag prior to correction and normalization. Panel B shows ratio data scaled by the 113 and 114 peak areas (for patient A and B respectively). Panel C shows ratio data after isotopic correction and median-based normalization.

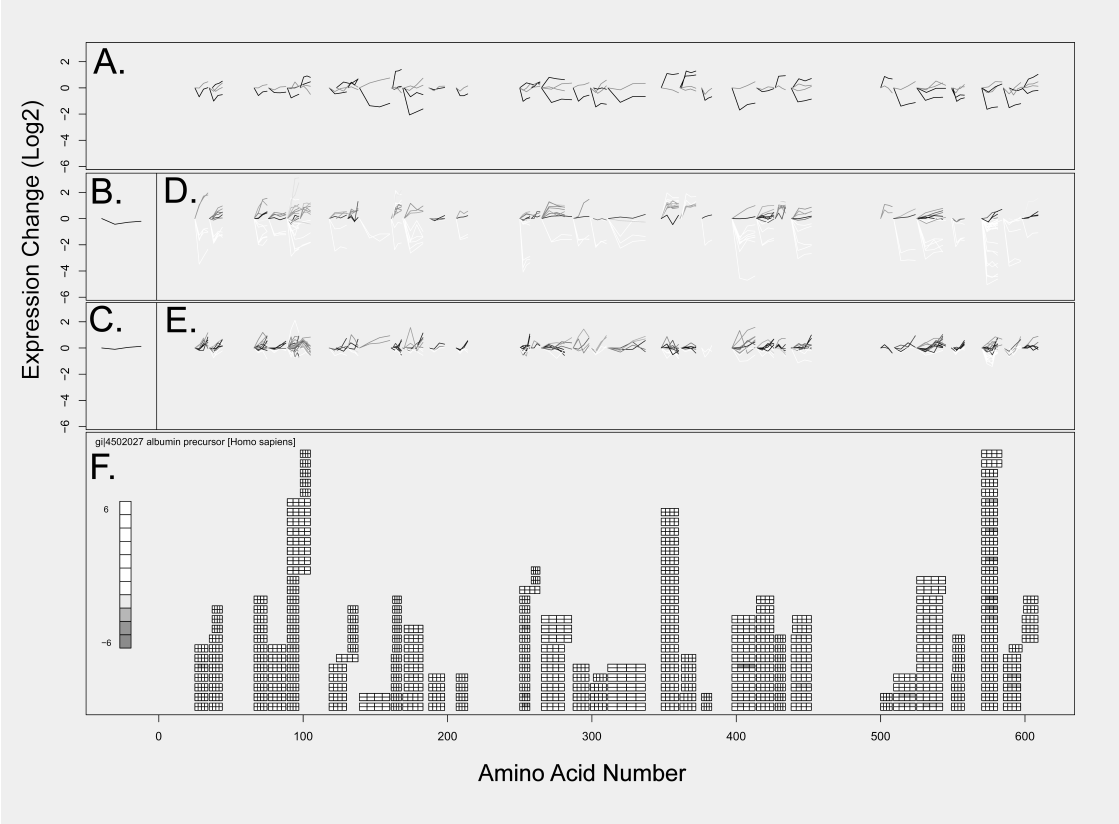


Figure 3.2 Peptide expression map generated by iTRAQPak for the albumin precursor protein (gi|4502027)

The map contains multiple views that present peptide expression values as either an expression plot (A-E) or heat-map (F). Expression values associated with each peptide are presented spatially within the context of the albumin precursor protein, where the x-axis indicates each peptide's location within the protein using amino acid numbers as coordinates. Expression values are condensed to either the peptide level (A) or protein level (B, C), or is uncondensed (D, E, and F). Expression data for both patients is also presented. Pane A contains expression plots for both patients and uses colored plot lines to discriminate between the two subjects, with black and red representing patients A and B respectively. Panes B and D represent subject A, and panes C and E represent subject B. The heat-maps in pane F present expression patterns for both subjects; Figure 3.3 describes this pane in more detail.

The heat-map was found to be an effective method for viewing longitudinal expression changes. For larger proteins (> 1200 AA) or very small peptides (<10 AA), it was found that plotting space was noticeably limited, causing peptide blocks to be somewhat compacted. It was possible to compensate for this by plotting the PDF output with wider page dimensions (> 13 inches) or by significantly increasing zoom factor. Both the peptide heat-maps and paired expression plots allowed expression profiles to be easily compared. Additionally, the SOM coloring allowed otherwise complex expression patterns to be more easily interpreted.

The importance of incorporating multilevel views into peptide expression plots was made evident after inspecting the heat-map and expression plots. Figure 3.3 shows several albumin peptides with several opposing expression patterns, especially for patient A. The N-terminal peptides (in the10-50 AA region), for example, shows an increase in expression over time (T1 to T2), while others show a strong decrease between T1 and T2. When averaged at the peptide level, these opposing expression trends are lost, favoring the decreased expression pattern. Figure 3.4 shows a more detailed view of the observed albumin peptides with data shown as expression plots for individual peptides. The expression change is plotted log base 2 such that increases in expression have positive slope and decreases in expression have a negative slope.

The *pepMod* function was applied to the expression data using parameters to highlight peptides lacking an N-terminus tag in a newly generated expression map. Shown in Figure 3.5 is a zoomed and cropped image of the albumin protein that was generated. Visual inspection of the output showed a number of highlighted peptides allowing us to determine which of the peptides shown in the heat-map were lacking an N-terminus tag. Interestingly, it was observed that many of the peptides lacking an

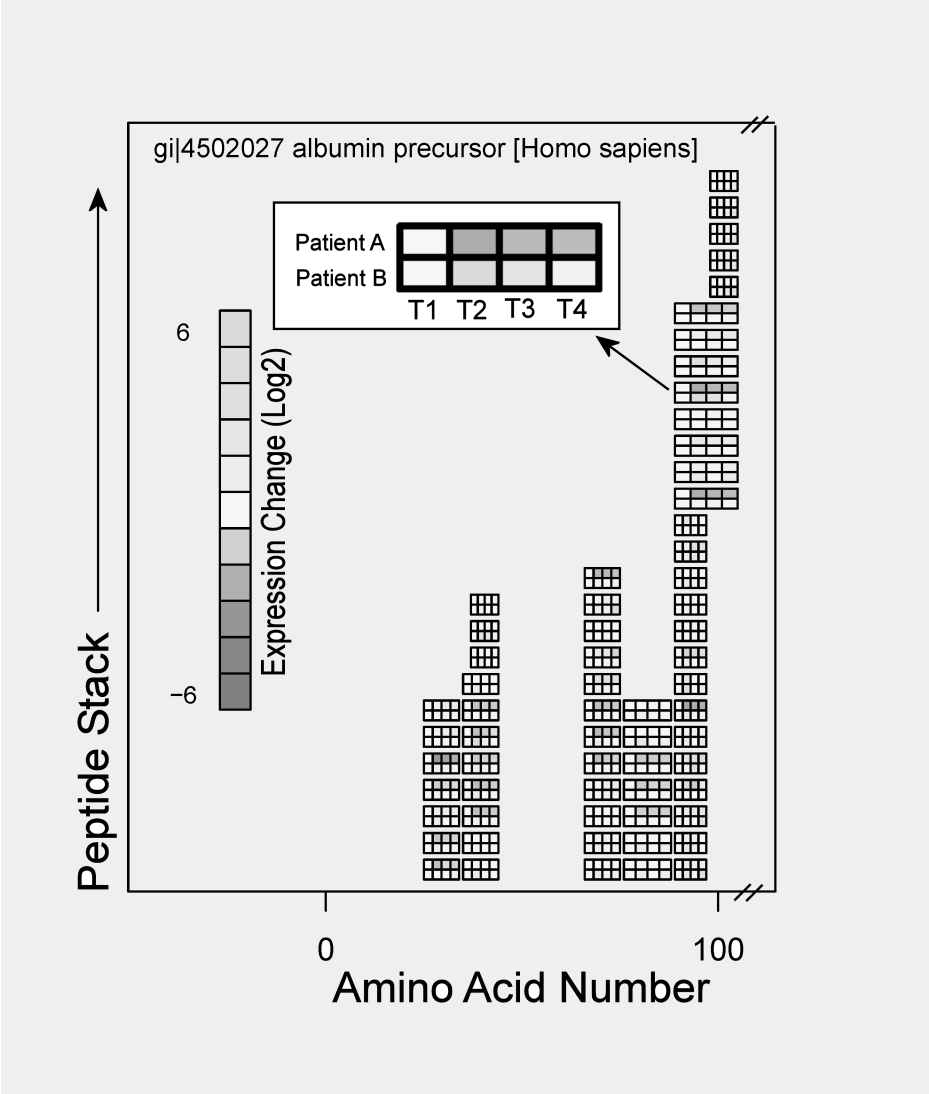


Figure 3.3 The iTRAQ analysis identified a number of peptides matching the albumin precursor protein amino acid sequence (gi|4502027)

In this zoom-in of the heat-map view (Pane F, Figure 3.2), the expression patterns for peptides corresponding to the first 120 amino acids of this protein are shown. Each rectangle represents a single peptide and contains 8 color graded boxes that indicate expression change; the top four boxes correspond to patient A, and the bottom four correspond to patient B. Columns represent the four time points, where expression changes are expressed as a scaled value relative to time 0. The magnitude of the change is determined by the color scale. Peptide rectangles are stacked in the direction of the y-axis (Peptide Stack) simply as a means of organizing data presentation.

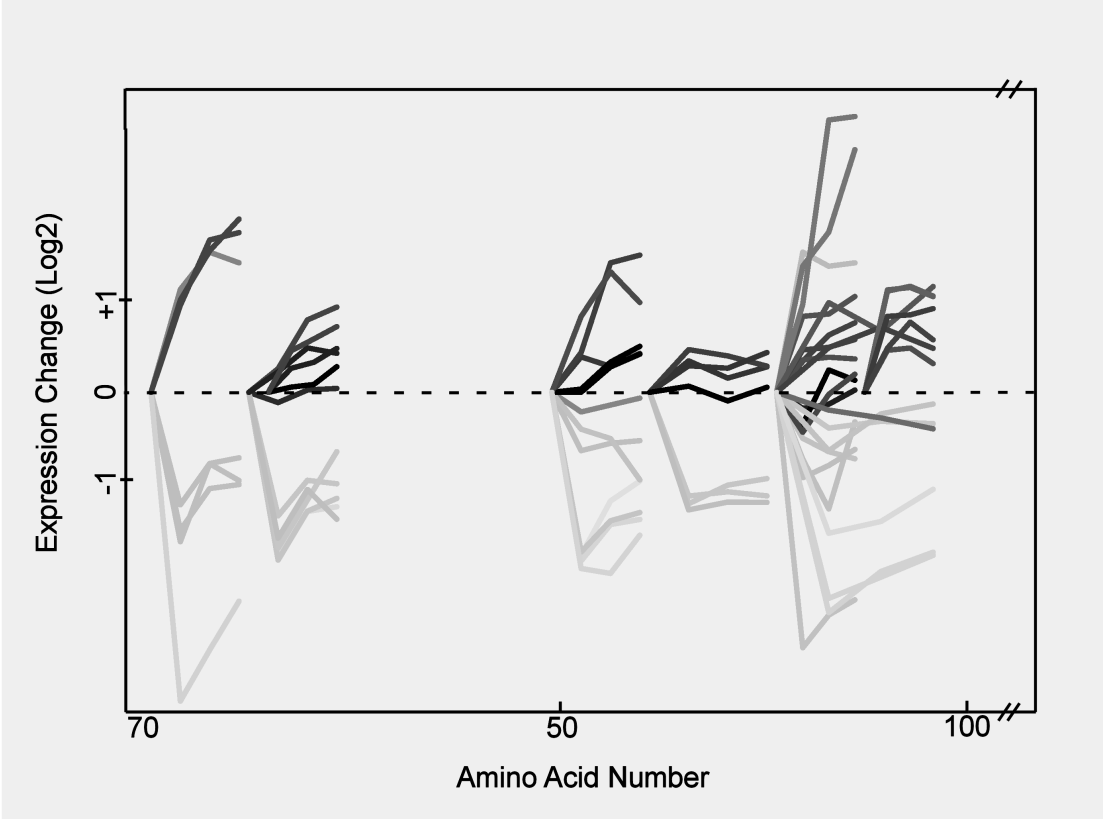


Figure 3.4 Expression plots for several peptides corresponding to the albumin precursor protein (gi|4502027)

Each plot line corresponds to the expression values of a single peptide and represents the expression change over time. The x-axis represents the relative location of the peptides within the context of the albumin precursor protein, shown here are only peptides that match the 70-100 amino acid region of the protein. A SOM coloring scheme is applied to each plot line to facilitate rapid visualization of similar expression trends; shown here in grayscale, similar shades indicate similar expression patterns.

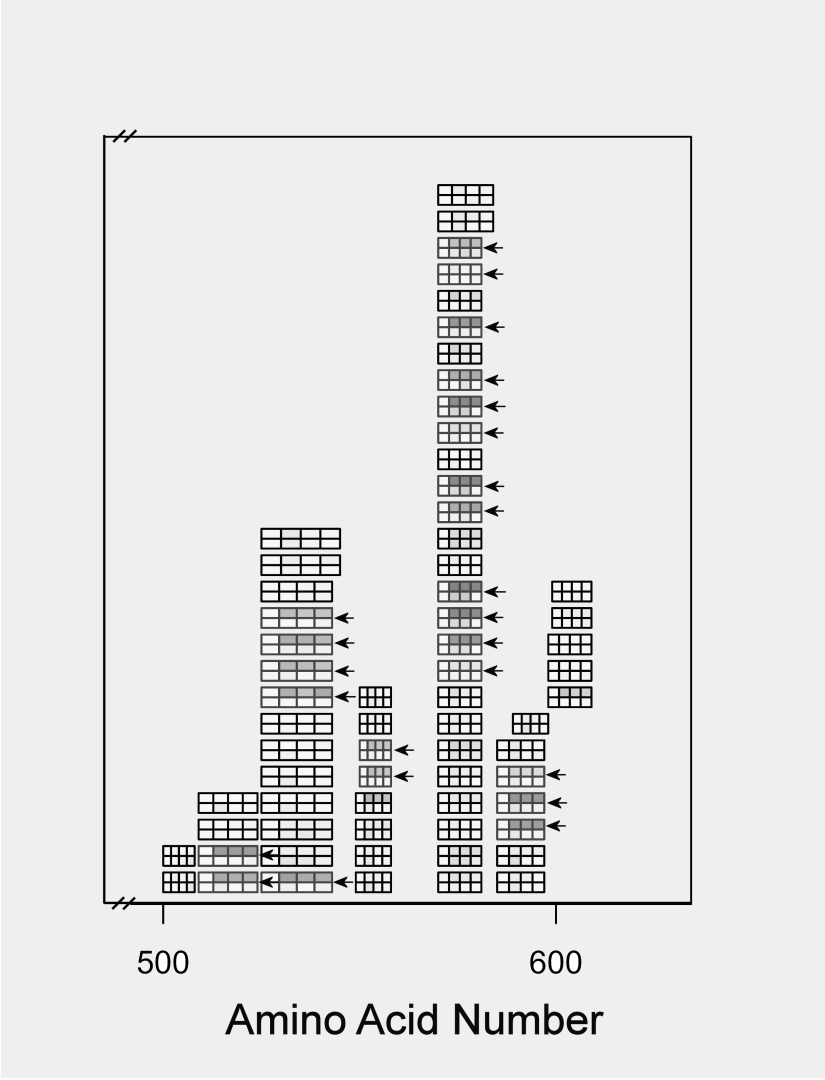


Figure 3.5 Modifications highlighting

Using the pepMod function, peptides containing modifications of interest may be highlighted. Peptides lacking an N terminal tag are highlighted in blue, shown here in grayscale, but further highlighted with an arrow.

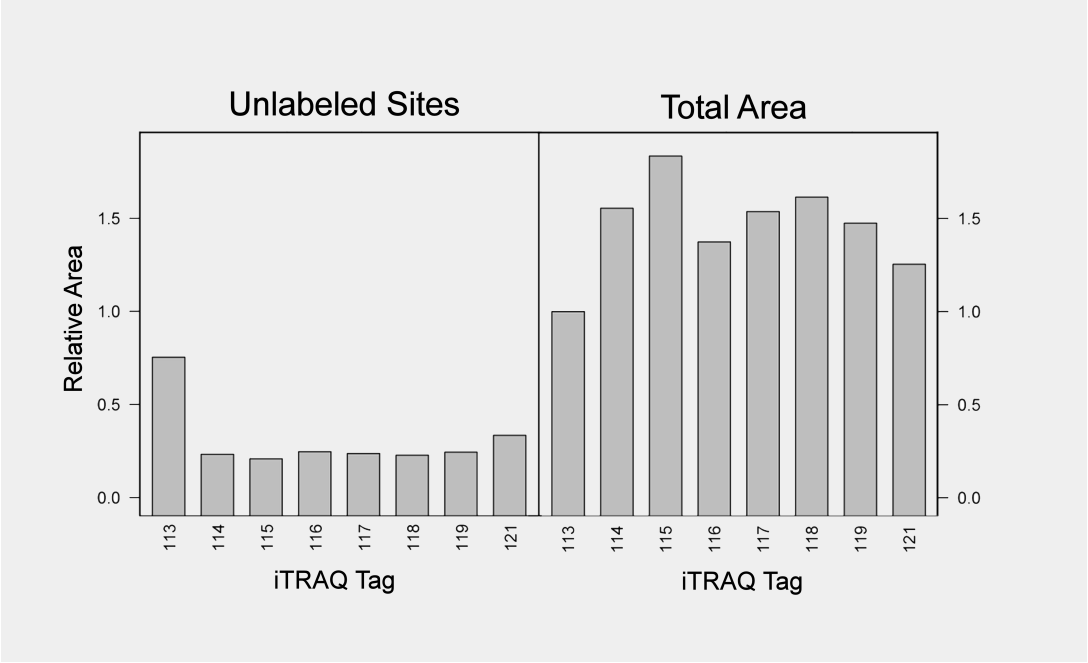


Figure 3.6 Quality control measures applied to shotgun data to assess labeling efficiency
Unlabeled sites, grouped by tag, are quantified as a function of the relative peak areas of partially and fully labeled peptides. Total area is a measure of total protein identified by isobaric tagging analysis and is quantified as a function of total peak area, grouped by tag.

N-terminus tag also showed a strong decrease in expression over time (0-9 months) for patient A, but this was not observed for patient B. To determine if this relationship was statistically significant, a three way ANOVA was applied to the expression data for the peptides highlighted in Figure 3.5. A highly significant interaction was found between the patient and time terms ($p < 0.0001$), and a highly significant three-way interaction ($p < 0.0001$) between the patient, time, and modification terms. These results show that the expression pattern differences between Subject A and Subject B are significantly correlated with the lack of an N-terminus tag. This strongly suggests that the different expression patterns seen between Subject A and Subject B are due to a labeling effect, rather than a biological effect.

To investigate this further, the *LabelingEfficiency* function was applied to the expression data set to determine which if any of the tags may be contributing the observed trends in Figure 3.5. The resulting plot from the application of this function is shown in Figure 3.6. The plot shows a noticeable difference in the relative area of unlabeled/labeled peptides associated with the 113 labeled sample as compared to the samples labeled by the other seven tags. From this, we can determine that there are relatively more partially labeled peptides in the 113 labeled sample as compared to the other seven samples, suggesting the 113 labeled sample was not as efficiently labeled as the other seven samples. Experimentally, all steps were performed similarly among the different samples and labels. However, this observation adds support to the possibility that the expression patterns observed in Figure 3.5 may be due in part to a labeling effect rather than a biological effect. Further, the observed drop in expression between 0 and 3 months, for patient A, in peptides lacking an N-terminus tag also supports the observation that unlabeled peptides are more abundantly represented in the 113 labeled sample.

3.5 Conclusion

We developed the iTRAQPak software package to aid in the visualization and interpretation of 8-plex shotgun proteomics expression data generated from a four time-point longitudinal study involving two patients. The isotope impurity and transformation functions available in this package facilitate rapid analysis of complex expression data. The iTRAQPak expression maps enable expression data to be viewed on multiple levels using methods that reduce visual complexity. We used the peptide expression maps to reveal complex patterns of expression between peptides identified as being fragments of same protein and even peptides with identical sequences. This observed complexity of expression further emphasizes the importance of interpreting condensed expression data with caution. Using the iTRAQPak modifications highlighting feature we observed a visual correlation among expression patterns from peptides lacking an N-terminal tag and also found this correlation to be statistically significant and

possibly related to the labeling efficiency of 113 isobaric tags in these experiments. While we have used this package's functionality to identify likely factors contributing to experimental variation within our data set, the approaches demonstrated here are equally applicable to the identification of expression trends that are of important biological significance. Finally, although iTRAQPak was developed for 8-plex data sets, it may be applied to 4-plex data sets as well. Data from electrospray based analyses could be incorporated if formatted appropriately. iTRAQPak is freely available for non-commercial purposes and can be downloaded from the Comprehensive R Archive Network (CRAN): http://cran.r-project.org/. This is most easily achieved from the R interface, first by selecting 'Install Packages' from the 'Packages' menu bar, and then selecting iTRAQPak from the packages list.

REFERENCES

1. Chen X, Sun LW, Yu YB *et al*. **Amino acid-coded tagging approaches in quantitative proteomics**. *Expert Review of Proteomics* (2007) 4:25-37.

2. Hattan SJ, Marchese J, Khainovski N *et al*. **Comparative study of [three] LC-MALDI workflows for the analysis of complex proteomic samples**. Journal of Proteome Research (2005) 4:1931-1941.

3. Choe LH, D'Ascenzo M, Relkin NR *et al*. **8-Plex Quantitation of changes in cerebrospinal fluid protein expression in subjects undergoing intravenous immunoglobulin treatment for Alzheimer's disease**. *Proteomics* (2007); in press.

4. Aggarwal K, Choe LH, Lee KH. **Quantitative analysis of protein expression using amine-specific isobaric tags in Escherichia coli cells expressing rhsA elements**. *Proteomics* (2005) 5:2297-2308.

5. D'Ascenzo M, Relkin NR, Lee KH. **Alzheimer's disease cerebrospinal fluid biomarker discovery: A proteomics approach**. *Current Opinion in Molecular Therapeutics* (2005) 7:557-564.

6. Gentleman R, Carey V, Huber W *et al*. **Bioinformatics and Computational Biology Solutions Using R and Bioconductor**. Springer, 2005.

7. Quackenbush J**. Microarray data normalization and transformation**. *Nature Genetics* (2002) 32:496-501.

8. Gilbert D, Schroeder M, van Helden J. **Interactive visualization and exploration of relationships between biological objects**. *Trends Biotechnol* (2000) 18:487-494.

CHAPTER 4

CONCLUDING REMARKS AND FUTURE DIRECTIONS

The pathology of AD is quite complex and we are only now beginning to untangle the nuances of this disease. AD has been inextricably linked with the buildup of β-amyloid plaques within brain tissue; however the ultimate cause of this buildup remains unclear. A variety of models have been proposed to explain this phenomena, proposing the role of a number of neurological mechanisms and responses, including blood brain barrier clearance and regulation mechanisms, the inflammation response, and the cell death response. However, none of these models fully explain the pathology and many questions remain.

To fully understand this complex disease, we must first identify key molecular constituents, be it genes, proteins, or inorganic molecules that are integral to its pathology. With the knowledge of these compounds in hand, we can begin to piece together their physiology and their underlying role in the pathology of the disease. In Chapter 1, the role of proteomics and the study of CSF was discussed and linked to this process of discovery. Tools such as 2DGE, MS, and high throughput methods such as iTRAQ are being used to study important aspects of CSF physiology that may further our insight into AD pathology. While these methods are not without their limitations, they offer an unprecedented level of detail, precision, and throughput for understanding molecular constituents. With these tools, we may be able to identify proteins integral to the complex pathology underlying AD. And with the high throughput nature of methods such as iTRAQ, it will be possible to sample larger populations of individuals in less time than previously possible. This higher throughput, in the presence of an adequate supply of biological samples, may allow us to gain a more complete picture of the AD pathology at the individual level and how it may vary within a larger population of individuals.

In the interim, while we come to more fully understand the molecular basis of this disease, there is still an immediate need for accurate diagnoses of AD in living individuals. In

addition, there is a need to monitor the progress of the disease in subjects that undergo treatment. With no current methods available to diagnose and monitor the progress of AD in living subjects with certainty, it is becoming increasingly important to identify key molecules integral to AD pathology that would make this possible. In the absence of a unifying model of AD pathology, we can to turn to biomarker discovery methods such as those described in Chapter 2 in an effort to identify these key proteins. These discovery methods do not require a full understanding of AD pathology and allow us to find correlation and potential biological patterns of interest in complex biological data sets that otherwise would be difficult, if not impossible, to detect.

4.1 Future Directions

Next steps for work described in Chapter 2 would be to obtain CSF specimens from a larger cohort of diseased and healthy individuals. While the number of subjects considered in the analyses in Chapter 2 represents a large population in relative terms, as compared other AD studies of its kind, drawing conclusions from a larger pool of subjects may increase our overall confidence in obtained results. Repeating similar analyses with a larger sample size, may improve the possibility of identifying biologically relevant proteins. Additionally, with larger cohorts, it may be possible to identify a natural organization of expression patterns due to subpopulations that may exist within diseased subjects, allowing multiple mechanisms of pathology to be discovered.

In Chapter 3, the iTRAQPak package was presented and its utility with iTRAQ 8-plex based protein expression data was demonstrated. Since its introduction to the research community in the spring of 2008, there has been an immediate interest in this package. This interest demonstrates the research community's need for such an application, and further, it enables correspondence with users to identify potentially useful features that could be incorporated into future development efforts. Correspondence with one user has revealed a desire to incorporate support for addition data import formats, beyond the currently supported

MascotGPS data format. Suggestions for alternate format support included output generated by the software application ProteinPilot™ (ABI) and a more universal XML based format, pepXML.

iTRAQPak was developed in the R programming language. While R is known to be a robust programming environment for the development of statistical and graphical applications, it can be somewhat daunting to those unfamiliar with this programming environment. In an effort to offer iTRAQPak's functionality in a more user friendly environment, future development could seek to incorporate elements of this package into a standalone environment based on programming languages such as JAVA or C++. Such languages have the advantage of being more application centric and allow rich graphical user interfaces (GUI) to be developed. Such an interface would allow data to be easily imported and saved in a manner that is familiar to most users. In addition, the interface would allow a more interactive exploration of experimental data. For example, protein expression maps, instead of being static PDF images, could be displayed to allow such functionality as zooming, feature highlighting, expression pattern searches, and pattern clustering. The environment could also allow data to be linked to web-based resources or local SQL databases for quick lookups of protein identities and function. Or, it could enable the storage of annotation and literature searches relating to expression profiles of interest.

APPENDIX

ITRAQPAK SOURCE CODE

```
LabelingEfficiency <-
function(
    file,
    outfile="Efficiency.pdf"
  )
{
  ## Load and transform data
  idata <-
    TransformData(
      LoadData(
        file,
        mod.eval=T
      ),
      norm.method=0,
      ratio=F,
      LOG=F
    )

  dat <- idata$DATA

  ## Select dat subsets based on N or K labeled tag
  datN1 <- dat[dat$N == 1,]
  datN2 <- dat[dat$N == 2,]
  datK1 <- dat[dat$K == 1,]
  datK2 <- dat[dat$K == 2,]

  ## Subset further
  datN1K1 <- datN1[datN1$K == 1,]
  datN1K2 <- datN1[datN1$K == 2,]
  datN2K1 <- datN2[datN2$K == 1,]

  ## Identify peptides that contain a K, but lack a K labeled tag
  datKK2 <- datN1K2[grep("K", paste(datN1K2$Best.Peptide.Sequence)),]

  ## Comine reshaped subsets into single data.frame
  fsets <-
    rbind(
      data.frame(
        reshape(datN1K1, idvar="id", varying=c(idata$COL.NAMES),
          direction="long"), grp="Peptide Class: A"
      ),
      data.frame(
        reshape(datN2K1, idvar="id", varying=c(idata$COL.NAMES),
          direction="long"), grp="Peptide Class: B1"
      )
    )

  fsets <-
    rbind(
      fsets,
      data.frame(
        reshape(datKK2, idvar="id", varying=c(idata$COL.NAMES),
          direction="long"), grp="Peptide Class: B"
      )
    )

  fsets <-
    rbind(
      fsets,
      data.frame(
        reshape(dat, idvar="id", varying=c(idata$COL.NAMES),
          direction="long"), grp="Peptide Class: All"
```

```
    )
  )

fsets$grp <- factor(fsets$grp)
fsets$time <- factor(fsets$time)

## Calculate total area of each subset
datKK2.sum <- colSums(datKK2[idata$COL.NAMES])
datN2K1.sum <- colSums(datN2K1[idata$COL.NAMES])
datN1K1.sum <- colSums(datN1K1[idata$COL.NAMES])
dat.sum <- colSums(dat[idata$COL.NAMES])

## Calulate efficiency ratio: (B1 + B2) / A
eff.rats <-
  data.frame(
    Area=(datN2K1.sum + datKK2.sum)/datN1K1.sum,
    id=names(datN2K1.sum),
    grp="(B1 + B2) / A"
  )

## Calulate efficiency ratio: B1 / A, bind to vals
eff.rats <-
  rbind(
    eff.rats,
    data.frame(
      Area=datN2K1.sum/datN1K1.sum,
      id=names(datN2K1.sum),
      grp="B1 / A"
    )
  )

## Calulate efficiency ratio: B2 / A, bind to vals
eff.rats <-
  rbind(
    eff.rats,
    data.frame(
      Area=datKK2.sum/datN1K1.sum,
      id=names(datKK2.sum),
      grp="B2 / A"
    )
  )

## Calculate sum of all peptide areas, scale to TAG[1]
eff.rats <-
  rbind(
    eff.rats,
    data.frame(
      Area = sumrat(dat.sum),
      id=names(dat.sum),
      grp="Total Area (Scaled to 113)"
    )
  )

eff.rats$grp <- factor(eff.rats$grp)

## Create PDF file, 4 plots per page
pdf(file=outfile, width=8, height=8)
par(mfrow=c(2, 2))

## Boxplots
bp <- bwplot(
  log2(Area)~time | grp,
  data=fsets,
  as.table=T,
  scales=list(x=list(rot=90))
)

print(update(bp, xlab="TAG"))

## Barchart lattice, plots 5-8
b <-
```

```
    barchart(
      Area~id | grp,
      data=eff.rats,
      col="grey",
      xlab="Ratio",
      horizontal=F,
      as.table=T,
      main=NULL,
      scales=list(x=list(labels=sub("Area.", "", levels(eff.rats$id)), rot=90))
    )

  print(update(b,xlab="TAG", ylab="%"))

  quiet <- dev.off()
}

LabelingEfficiencyXY <-
function(
    file,
    outfile="EfficiencyXY.pdf"
  )
{
  col.fun <-
    expression(
      apply(
        X,
        MARGIN=1,
        FUN = function(x){
          if(
            (x["K"]==1 & x["N"]==2) |
            (x["N"]==1 & x["K"]==2 &
            any(grep("K", x["Best.Peptide.Sequence"])))
          ) "red" else "black"
        }
      )
    )

  TagComparisonPlots(file=file, col.fun=col.fun, outfile=outfile)
}

LoadData <-
function(
    file,
    rm.na=T,
    mod.eval=F,
    sep="\t",
    tags.s1=c(113, 115, 117, 119),
    tags.s2=c(114, 116, 118, 121)
  )
{
  ## Assign columns to S1 and S2 given tag parameters
  cols.s1 <-
    sapply(
      tags.s1,
      FUN=function(x) paste("Area", x, sep=".")
    )

  cols.s2 <-
    sapply(
      tags.s2,
      FUN=function(x) paste("Area", x, sep=".")
    )

  ## Read the data
  dat <- read.table(file=file, sep=sep, header=T)

  ## Remove rows w/ NA values
  if(rm.na == T){
    dat <- dat[complete.cases(dat),]
  }
```

```
  ## Rename lengthy column names
  names(dat)[names(dat) == "Start.Sequence.Position"] <- "Start"
  names(dat)[names(dat) == "End.Sequence.Position"] <- "Stop"

  ## Create unique id column
  dat <-
    cbind(
      id = c(1:dim(dat)[1]),
      dat
    )

  ## Order by start and stop
  dat <- dat[order(dat$Start, dat$Stop),]

  ## Evaluate secondary modifcations
  if(mod.eval == T){
    dat <- addModCols(dat)
  }

  ## Set default expression heatmap highlight to black
  dat$mod.col = 'black'


  ## Create idata
  idata <-
    list(
      TAGS.S1 = tags.s1,
      TAGS.S2 = tags.s2,
      COLS.S1 = cols.s1,
      COLS.S2 = cols.s2,
      COL.NAMES = c(cols.s1, cols.s2),
      DATA = dat
    )

  return(idata)
}

PepMod <-
function(
       idata,
       color="blue",
       mod.ids=c("K","Y","N-term")
     ){

  d <- idata$DATA

  pep.cols <-
    sapply(1:dim(d)[1],
      function(x)
        if(unlabeledSiteCount(
              d[x,"Best.Peptide.Sequence"],
              d[x,"Modification"],
              mod.ids=mod.ids
           ) > 0)
          return("blue")
        else
          return("black")
    )

  idata$DATA$mod.col <- pep.cols

  return(idata)
}

PlotExpression <-
function(
    file=NULL,
    idata=NULL,
    outfile="gis.pdf",
    gis="all",
    sig.table=NULL,
```

```
    som.xdim=12,
    som.ydim=15,
    topol="rect",
    neigh="bubble",
    radius=3,
    alpha=0.05,
    match.types=0,
    trend=NULL,
    pwidth=15,
    append.output=T
  )
{

  ## Check Trend Input
  if(any(2 == match.types) & is.null(trend)){
    e <- simpleError(
          paste("SOM trend matching specified without trend object.",
                "(match.types=2, trend=NULL) "))
    stop(e)
  }else if(any(3 == match.types) & is.null(trend)){
    e <- simpleError(
          paste("Trend matching specified without trend object.",
                "(match.types=3, trend=NULL) "))
    stop(e)
  }

  ## Load and transform data to package default
  if(is.null(idata)){
    idata <- TransformData(LoadData(file))
  }

  ## Count number of gis
  if(gis[1] == 'all'){
    gis <- levels(idata$DATA$Accession.Number)
  }else{
    gis <- levels(factor(gis))
  }

  ## Train SOM
  som.train <- trainSom(idata, som.xdim, som.ydim)

  ## Draw maps, one for each GI
  for(i in 1:length(gis)){

    if(sum(match.types) > 0){
      match.types <- c(0, match.types)
    }

    ## Assign output name, conditionally
    if(append.output == F){
      outfile = paste(sub("\\|", ".", gis[i]), ".pdf", sep="")
    }

    ## Make a map for each match.type
    for(match.type in match.types){

      if(names(dev.cur()) != "pdf"){
        pdf(file=outfile, height=10, width=pwidth, pointsize=12)
      }

      draw(
        idata,
        gi=gis[i],
        som.train=som.train,
        match.type=match.type,
        sig.table=sig.table,
        pwidth=pwidth,
        trend=trend
      )

      ## Close the outfile if not appending, suppress output
```

```
      if(append.output==F){
        quiet <- dev.off()
      }
    }
  }

  ## Insure pdf device is closed
  if(names(dev.cur()) == "pdf"){
    quiet <- dev.off()
  }
}

TagComparisonPlot <-
function(
    idata,
    x.col,
    y.col,
    col=1,
    main
  )
{

  dat <- idata$DATA

  xlab <- x.col
  ylab <- y.col
  dat.xy <- cbind(dat[, x.col], dat[, y.col])
  xy.max <- max(dat.xy)
  xy.min <- min(dat.xy)

  plot(
    dat[, x.col],
    dat[, y.col],
    las=2,
    main = main,
    cex.main=.85,
    xlim = c(xy.min, xy.max),
    ylim = c(xy.min, xy.max),
    col = col,
    xlab = xlab,
    ylab = ylab
  )

  xy.max.c <- ceiling(xy.max)
  xy.min.f <- floor(xy.min)
  lines(c(xy.min.f, xy.max.c), c(xy.min.f, xy.max.c))
}

TagComparisonPlots <-
function(
    file=NULL,
    idata=NULL,
    rm.na=T,
    sep="\t",
    correct=F,
    norm.method=0,
    LOG=T,
    col="black",
    col.fun=NULL,
    outfile="TagComparison.pdf",
    tags.s1=c(113, 115, 117, 119),
    tags.s2=c(114, 116, 118, 121)
  )
{

  ## Load and transform data
  if(!is.null(file) & is.null(idata)){
    idata <-
      TransformData(
        LoadData(
          file,
```

```
          rm.na=T,
          mod.eval=T,
          sep="\t",
          tags.s1=tags.s1,
          tags.s2=tags.s2
        ),
        correct=correct,
        norm.method=norm.method,
        ratio=F,
        LOG=T
      )
  }

  ## Apply coloring via function if supplied
  if(!is.null(col.fun)){
    X <- idata$DATA
    col <- eval(expr=col.fun, envir=X)
  }

  ## Plot images to PDF
  pdf(outfile, height=8, width=8)

  for(j in 1:2){
    par(mfrow=c(2, 2))
    if(j == 1) cols <- idata$COLS.S1 else cols <- idata$COLS.S2
    for(i in 1:4){
     TagComparisonPlot(
        idata,
        cols[1],
        cols[i],
        col=col,
        paste(cols[1], cols[i], sep=" x ")
      )
    }
  }

  quiet <- dev.off()
}

TagSummaryBoxPlot <-
function(file, sep="\t", outfile="TagSummary.pdf"){

  idata <-
    LoadData(
      file,
      rm.na=T,
      mod.eval=F,
      sep=sep
    )

  pdf(file=outfile, height=10, width=14, pointsize=12)

  ## log2, uncorrected unnormalized raw
  idata1 <- TransformData(idata, correct=F, LOG2=T, ratio=F, norm.method=0)

  ## log2, uncorrected unnormalized ratio
  idata2 <-
    TransformData(idata, correct=F, LOG2=T, ratio=T, norm.method=0)

  ## log2, corrected median normalized ratio
  idata3 <-
    TransformData(idata, correct=T, LOG2=T, ratio=T, norm.method=1)

  ## log2, corrected mean normalized ratio
  idata4 <-
    TransformData(idata, correct=T, LOG2=T, ratio=T, norm.method=2)

  ## log2, corrected lowess normalized ratio
  idata5 <-
    TransformData(idata, correct=T, LOG2=T, ratio=T, norm.method=3)
```

```
  ## log2, corrected lowess normalized ratio
  idata6 <-
    TransformData(idata, correct=T, LOG2=F, ratio=F, norm.method=3)

  ## Generate BoxPlots
  par(mfrow=c(2, 3))

  boxplot(
    idata1$DATA[,idata1$COL.NAMES],
    las=2,
    main = "Uncorrected, Unnormalized, LOG2",
    cex.main=.85
  )

  boxplot(
    idata2$DATA[,idata2$COL.NAMES],
    las=2,
    main = "Uncorrected, Unnormalized, Ratio, LOG2",
    cex.main=.85
  )

  boxplot(
    idata3$DATA[,idata3$COL.NAMES],
    las=2,
    main = "Corrected, Median Normalized, Ratio, LOG2",
    cex.main=.85
  )

  boxplot(
    idata4$DATA[,idata4$COL.NAMES],
    las=2,
    main = "Corrected, Mean Normalized, Ratio, LOG2",
    cex.main=.85
  )

  boxplot(
    idata5$DATA[,idata5$COL.NAMES],
    las=2,
    main = "Corrected, LOWESS Normalized, Ratio, LOG2",
    cex.main=.85
  )

  quiet <- dev.off()
}

TransformData <-
function(
    idata,
    correct=T,
    LOG2=T,
    ratio=T,
    norm.method=1,
    dat.subset=NULL,
    rm.na=T,
    cf.table=NULL
  )
{

  ## Correct for carryover between peaks
  if(correct){

    ## Use default correction values if cf.table is NULL
    if(!is.null(cf.table)){
      CF.TABLE <- cf.table
    }

    ## Correct area values
    idata$DATA[,idata$COL.NAMES] <-
      correct(
        idata$DATA[,idata$COL.NAMES],
        c(idata$ISO.S1, idata$ISO.S2),
```

```
        CF.TABLE
      )
  }

  ## [OPTION 1]  Median, all to S1 (t0)
  if(norm.method == 1){
    idata$DATA[,idata$COL.NAMES] <-
      normMedian(idata$DATA[,idata$COL.NAMES])
  ## [OPTION 2]  Mean, all to S1 (t0)
  }else if(norm.method == 2){
    idata$DATA[,idata$COL.NAMES] <-
      normMean(idata$DATA[,idata$COL.NAMES])
  ## [OPTION 3]  LOWESS
  }else if(norm.method == 3){
    idata$DATA[,idata$COL.NAMES] <-
      normLowess(idata$DATA[,idata$COL.NAMES])
  ## [OPTION 4]  Median, to S1 and S2 (t0)
  }else if(norm.method == 4){
    idata$DATA[,idata$COLS.S1] <-
      normMedian(idata$DATA[,idata$COLS.S1])
    idata$DATA[,idata$COLS.S2] <-
      normMedian(idata$DATA[,idata$COLS.S2])
  ## [OPTION 5]  Median, to subset, to S1 (t0)
  }else if(norm.method == 5){
    idata$DATA[,idata$COL.NAMES] <-
      normMedianSubset(
        idata$DATA[,idata$COL.NAMES],
        dat.subset[,idata$COL.NAMES]
      )
  }

  ## RATIO
  if(ratio){
    idata$DATA[,idata$COLS.S1] <- ratio(idata$DATA[,idata$COLS.S1])
    idata$DATA[,idata$COLS.S2] <- ratio(idata$DATA[,idata$COLS.S2])
  }

  ## LOG2 Transformation
  if(LOG2){
    idata$DATA[,idata$COL.NAMES] <- log2(idata$DATA[,idata$COL.NAMES])
  }

  ## Remove NA values
  #dat <- dat[complete.cases(dat),]
  if(rm.na){
    idata$DATA  <- IDPmisc::NaRV.omit(idata$DATA)
  }

  return(idata)
}

addModCols <-
function(dat){

  dat[grep("Oxidation", paste(dat$Modification)), "O"] <- 1
  dat[grep("MMTS", paste(dat$Modification)), "M"] <- 1
  dat[grep("\\(N-term\\)\\[0\\]", paste(dat$Modification)), "N"] <- 1
  dat[grep("8plex \\(K\\)", paste(dat$Modification)), "K"] <- 1
  dat[grep("8plex \\(Y\\)", paste(dat$Modification)), "Y"] <- 1

  if(length(dat[is.na(dat)]) > 1){
    dat[is.na(dat)] <- 2
  }

  return(dat)
}

addSigValues <-
function(sc, sig.table, col.name, ymin.global){

  peptides <- levels(factor(sc$peptide))
```

```
  for(i in 1:length(peptides)){

    peptide <- peptides[i]

    sigVal <- sig.table[sig.table$Peptide == peptide, col.name][1]

    if(!is.na(sigVal)){

      x <- sc[sc$peptide == peptide, "x"][1]

      text(x, ymin.global + 0.5, format(sigVal, digits=4), cex=.7,
           pos=4, srt=90)
    }
  }
}

as.trend <-
function(s){
  return(parse(text=gsub("[Tt]", "V", x=s)))
}

colorize <-
function(x){

  ## Set.max.val, let min max.val = 1
  max.val <- ceiling(max(abs(x)))

  y <- sapply(
        x,
        FUN = function(x)
                ifelse(
                  x > 0,
                  (x/max.val) * 100 / 2 + 50,
                  (1-abs(x/max.val)) * 100 / 2
                )
        )

  ## Zero not a color
  ## Let zero values = 0.001
  y[y==0] <- 0.001

  return(y)
}

correct <-
function(x, tags, cf.table){

  for(i in tags){

    ## Assign column name
    col.name <- paste("Area.", i, sep="")

    ## Get corection factor
    cf <- (correctionFactor(i, cf.table)/100)

    x[, col.name] <-  x[, col.name] - (x[,col.name] * cf)
  }

  return(x)
}

correctionFactor <-
function(tag, cf.table){

  tags <- c(113, 114, 115, 116, 117, 118, 119, 121)
  correction.factors <- cf.table

  cf = 0

  for(i in c(-2,-1,1,2)){
```

```
    if(tag - i >= min(tags) & tag + i <= max(tags)){
      cf.val <- correction.factors[paste(tag + i), getCol(i)]
      if(!is.na(cf.val)){
        cf <- (cf + cf.val)
      }
    }
  }
  return(cf)
}

draw <-
function(
    idata,
    gi,
    som.train,
    match.type,
    sig.table,
    pwidth,
    trend
  )
{

  ## Get DATA from idata
  dat <- idata$DATA

  ## Set output ajustment parameters for plots
  vadj.d <- -0.4
  vadj.a <- .273

  palette("default")

  ## Get the protein name
  name <- paste(dat[dat$Accession.Number == gi, "Protein.Name"][1])

  ## Draw plots
  drawBasePlot(gi, name)
  drawLines(dat, gi, som.train, match.type, sig.table, idata$COLS.S1,
            idata$COLS.S2, idata$COL.NAMES, trend, pwidth)
  drawAverageLines(dat, gi, vadj.a, sig.table, idata$COL.NAMES, pwidth)
  drawPeptides(dat, gi, idata$COL.NAMES, pwidth)
  drawAverageAllLines(dat, gi, idata$COLS.S1, idata$COL.NAMES, 1.78)
  drawAverageAllLines(dat, gi, idata$COLS.S2, idata$COL.NAMES, 1.295)
}

drawAverageAllLines <-
function(dat, gi, cols.subj, col.names, vadj){

  #get Y min/max .global
  ymin.global <- min(dat[,col.names])
  ymax.global <- max(dat[,col.names])

  #get only row w/ gi of interest
  dat <- dat[dat$Accession.Number == gi,]

  #calculate mean of each timepoint
  dat.m <-
    sapply(
      cols.subj,
      FUN = function(x) apply(t(dat[,x]), 1, mean)
    )

  #make plotable
  sc <- data.frame(x=c(1,2,3,4), y=dat.m)

  subplot(
    plot(
      sc$x,
      sc$y,
      #ylim = c(min(sc$y), max(sc$y)),
      ylim = c(ymin.global, ymax.global),
```

```
      xlim = c(0, 5),
      type="l",
      ylab = "",
      xlab = "",
      xaxt = "n",
      yaxt = "n",
      cex.axis = .75
    ),
    -.085,
    vadj,
    size=c(1, 1.8),
    vadj=1,
    hadj=0
  )
}

drawAverageLines <-
function(dat, gi, vadj, sig.table, col.names, pwidth){

  ## Get Y min/max .global
  ymin.global <- min(dat[,col.names])
  ymax.global <- max(dat[,col.names])

  ## Get only row with GI of interest
  dat <- dat[dat$Accession.Number == gi,]
  dat$Best.Peptide.Sequence <-
    dat$Best.Peptide.Sequence[drop = TRUE]

  dat <-
    cbind(
      dat,
      id2=factor(as.numeric(paste(dat$Start, dat$Stop, sep="."))),
      ordered=T
    )

  dat.f1<-
    sapply(
      col.names,
      FUN = function(x) tapply(dat[,x], factor(dat$id2), mean)
    )

  dat.coords.start <-
    sapply(
      unique(factor(dat$id2)),
      FUN = function(x) dat[dat$id2 == x , "Start"][1]
    )

  dat.coords.stop <-
    sapply(
      unique(factor(dat$id2)),
      FUN = function(x) dat[dat$id2 == x , "Stop"][1]
    )

  dat.peptides <-
    sapply(
      unique(factor(dat$id2)),
      FUN = function(x) paste(dat[dat$id2 == x , "Best.Peptide.Sequence"][1])
  )

  dat.f2 <-
    data.frame(
      Start=dat.coords.start,
      Stop=dat.coords.stop,
      t = rbind(dat.f1),
      peptide = dat.peptides
    )

  sc.1 <-
    getLines(
      dat.f2[,"Start"],
      dat.f2[,"Stop"],
```

```
      dat.f2[,3:6],
      0,
      rep(1, dim(dat.f2)[1]),
      peptide=dat.f2[,"peptide"]
    )

  sc.2 <-
    getLines(
      dat.f2[,"Start"],
      dat.f2[,"Stop"],
      dat.f2[,7:10],
      0,
      rep(1, dim(dat.f2)[1]),
      peptide=dat.f2[,"peptide"]
    )

  ## Set X max
  if(max(sc.1$x) <= 300)
    xmax.local <- 300
  else
    xmax.local <- max(sc.1$x)

  ## Set color palette
  palette("default")

  sp.a <- subplot(
    plot(
      0,
      0,
      ylim = c(ymin.global, ymax.global),
      xlim = c(-25, xmax.local),
      type="n",
      ylab = "Average Ratio",
      xlab = "",
      xaxt = "n",
    ),
    -.085,
    2.4,
    size=c(pwidth -1, 1.8),
    vadj=1 + vadj,
    hadj=0
  )

  ## Prepare graphics, plot lines within subplot sp.l
  op <- par(no.readonly=TRUE)
  par(sp.a)

  for(i in 1:nlevels(factor(sc.1$group))){
    lines(sc.1[sc.1$group==i,]$x, sc.1[sc.1$group==i,]$y, col=1)
    lines(sc.2[sc.2$group==i,]$x, sc.2[sc.2$group==i,]$y, col=2)
  }

  if(!is.null(sig.table))
    addSigValues(sc.1, sig.table, "Both", ymin.global)

  par(op)
}

drawBasePlot <-
function(gi, name){

  plot(
    0,
    0,
    xlim=c(0,2),
    ylim=c(0,2),
    frame.plot=F,
    yaxt = "n",
    xaxt = "n",
    ylab="",
    xlab="",
```

```
    type="n"
  )

  text(-.08,.77, paste(gi, name, sep=" "), pos=4, cex=.85)
}

drawColorScale <-
function(pos.x, pos.y, min.exp, max.exp){

  min.exp <- format(min.exp, digits=2)
  max.exp <- format(max.exp, digits=2)

  y <- c(0:10)/20
  color <- c(0:10) * 10 + .01

  rect(
    pos.x,
    pos.y + y,
    pos.x + 0.025,
    pos.y + y + 0.05,
    col=color
  )

  ## Add scale labels
  text(pos.x -0.03, pos.y + 0.02, min.exp, cex=0.75)
  text(pos.x -0.03, pos.y + 0.02 + .5, max.exp, cex=0.75)
}

drawLines <-
function(
    dat,
    gi,
    som.train,
    match.type=0,
    sig.table,
    cols.s1,
    cols.s2,
    col.names,
    trend,
    pwidth
  )
{

  som.colors.1 <- NULL
  som.colors.2 <- NULL

  ## Current default: set min max based on protein
  global.color.scale = F

  ## Get only row w/ gi of interest
  datg <- dat[dat$Accession.Number == gi,]

  ## Set color scale Y min/max .global
  ymin.global <- min(dat[,col.names])
  ymax.global <- max(dat[,col.names])

  ## Get the SOM, if there is more than one peptide
  if(dim(datg)[1] > 1){
    r.som.1 <- testSom(som.train, exp.vals = datg[,cols.s1])
    r.som.2 <- testSom(som.train, exp.vals = datg[,cols.s2])
    som.id.1 <- r.som.1$id
    som.id.2 <- r.som.2$id
  }else{
    som.id.1 <- 1
    som.id.2 <- 1
  }

  ## Set SOM color palette
  ## Cacluate color range from size of SOM grid
  ## Palette(rainbow(100, start=0, end=2/6))
  color.range <- som.train$xdim * som.train$ydim
```

```
palette(rainbow(color.range, start=.18, end=.15))

sc <-
  getLines2(
    datg[,"id"],
    datg[,"Start"],
    datg[,"Stop"],
    datg[,col.names],
    som.id.1,
    som.id.2,
    peptide=datg[,"Best.Peptide.Sequence"]
  )

## Determine match.type, get trends accordingly
if(match.type == 1)         trends <- NULL
else if(match.type == 2)    trends <- getSomTrends(som.train, trend)
else if(match.type == 3)    trends <- findTrends(datg, trend, cols.s1, cols.s2)



## Highlight matches
if(match.type > 0){
  #write.table(trends, file="trends2.txt")
  sc <- highlightLines(sc, match.type, trends)
}

## Set X max
if(max(sc$x) <= 300)
  xmax.local <- 300
else
  xmax.local <- max(sc$x)

## Plot Subject 1 (upper plot)
sp.u <- subplot(
  plot(
    0,
    0,
    ylim = c(ymin.global, ymax.global),
    xlim = c(-25, xmax.local),
    type="n",
    ylab = "Peak Area (log2 ratio)",
    xlab = "",
    xaxt = "n",
  ),
  -.085,
  1.78,
  size=c(pwidth-1, 1.8),
  vadj=1,
  hadj=0
)

## Plot Subject 2 (lower plot)
sp.l <- subplot(
  plot(
    0,
    0,
    ylim = c(ymin.global, ymax.global),
    xlim = c(-25, xmax.local),
    type="n",
    ylab = "Peak Area (log2 ratio)",
    xlab = "",
    xaxt = "n",
  ),
  -.085,
  1.295,
  size=c(pwidth-1, 1.8),
  vadj=1,
  hadj=0
)

## Prepare graphics, plot lines within subplot sp.l
```

```
  op <- par(no.readonly=TRUE)

  ## Order so colored highlights are on top
  sc <- sc[order(sc$s1.hl),]

  ## Plot lines for Subject 1
  for(i in unique(factor(sc$group))){

    col.s1 <- sc[sc$group==i, "s1.id"][1]

    if(match.type > 0 && sc[sc$group==i, "s1.hl"][1] == FALSE)
      col.s1 <- "grey"

    ## Plot Subject 1 (upper plot)
    par(sp.u)
    lines(
      sc[sc$group==i,]$x,
      sc[sc$group==i,]$y.s1,
      col=paste(col.s1)
    )
  }

  ## Add significance values if supplied
  if(!is.null(sig.table))
      addSigValues(sc, sig.table, "S1", ymin.global)

  sc <- sc[order(sc$s2.hl),]

  ## Plot lines for Subject  2
  for(i in unique(factor(sc$group))){

    col.s2 <- sc[sc$group==i, "s2.id"][1]

    if(match.type > 0 && sc[sc$group==i, "s2.hl"][1] == FALSE)
      col.s2 <- "grey"

    ## Lower plot
    par(sp.l)
    lines(
      sc[sc$group==i,]$x,
      sc[sc$group==i,]$y.s2,
      col=paste(col.s2)
    )
  }

  if(!is.null(sig.table))
    addSigValues(sc, sig.table, "S2", ymin.global)

  par(op)
}

drawPeptides <-
function(dat, gi, col.names, pwidth){

  ## Default level
  level.base <- .15
  level <- level.base
  max.levels <- 0

  ## Set color palette (red, yellow, green)
  palette(rainbow(100, start=0, end=2/6))

  dat.gi <- dat[dat$Accession.Number == gi,]
  dat.colors = dat.gi
  dat.colors[,col.names] <- colorize(dat.gi[,col.names])

  ## Set X max
  if(max(dat.gi$Stop) <= 300)
    xmax.local <- 300
  else
    xmax.local <- max(dat.gi$Stop)
```

```
  ## Set max pep levels
  max.pep.levels <- getMaxPepLevels(dat.gi)

  ## Create peptide plot area
  sp.p <- subplot(
    plot(
      0,
      0,
      ylim = c(.25, 11),
      xlim = c(-25, xmax.local),
      type="n",
      ylab = "",
      xlab = "",
      yaxt = "n",
      omi = c(0,0,0,0)
    ),
    -.085,
    -.25,
    size=c(pwidth-1, 4),
    vadj=0,
    hadj=0
  )

  ## Prepare graphics, plot lines within subplot sp.p
  op <- par(no.readonly=TRUE)
  par(sp.p)

  for(i in 1:dim(dat.gi)[1]){

    ## Increment rectangle levels, avoid overlap
    if(overlaps(dat.gi, i)){
      level <- (level + .4)
    }else{
      level <- level.base
    }

    r <-
      getRectangle(
        dat.gi[i,"Start"],
        dat.gi[i,"Stop"],
        level,
        as.vector(dat.colors[i, col.names], mode="numeric"),
        dat.gi[i, "mod.col"]
      )

    rect(r$xl, r$yb, r$xr, r$yt, col=r$exp.vals, border=paste(r$mod.col))
  }

  par(op)

  ## Draw color scale, assumes min/max set
  ## relative to protein, not global,
  ## see colorize function
  max.val <- ceiling(max(abs(dat.gi[,col.names])))
  drawColorScale(-.01, 0, -1 * max.val, max.val)
}

findTrends <-
function(dat, trend, cols.s1, cols.s2){

  #Rename to generic column names: Subject 1
  x <- dat[c(cols.s1, cols.s2)]
  names(x) <- rep(c("V1", "V2", "V3", "V4"), 2)

  #Evauluate trends
  s1 <- row.names(x[eval(trend, envir=x[1:4]),])
  s2 <- row.names(x[eval(trend, envir=x[5:8]),])

  y <- list(s1=as.vector(c(s1)), s2=as.vector(c(s2)))
```

```
  return(y)
}

getCol <-
function(x){
  if(x == -2)       return(4)
  else if(x == -1)  return(3)
  else if(x == 1)   return(2)
  else if(x == 2)   return(1)
}

getLines <-
function(start, stop, exp.vals, yshift, som.colors, peptide){

  x1 <-
    apply(
      array(1:3),
      1,
      FUN = function(x) start + ((stop - start)/3) * x
    )
  x1 <- cbind(start, rbind(x1))
  x2 <- data.frame(NULL)

  for(i in 1:length(start)){

    e1 <- data.frame(t(exp.vals[i,]))
    e1 <- e1 + yshift
    names(e1)[1]<- "y"
    x2 <-
      rbind(
        x2,
        data.frame(
          group=i,
          x=x1[i,],
          e1,
          somColor=som.colors[i],
          peptide[i]
        )
      )
  }

  row.names(x2) <- c(1:dim(x2)[1])

  return(x2)
}

getLines2 <-
function(
    ids,
    start,
    stop,
    exp.vals,
    s1.ids,
    s2.ids,
    peptide
  )
{

  ## Create start/stop array
  x1 <-
    apply(
      array(1:3),
      1,
      FUN = function(x) start + ((stop - start)/3) * x
    )
  x1 <- cbind(start, rbind(x1))
  x <- data.frame(NULL)

  for(i in 1:length(start)){

    s1.exp <- data.frame(t(exp.vals[i,1:4]))
```

```
    s2.exp <- data.frame(t(exp.vals[i,5:8]))
    names(s1.exp)[1]<- "y.s1"
    names(s2.exp)[1]<- "y.s2"

    ## Set line features, note hl (highlight) default = T
    x <-
      rbind(
        x,
        data.frame(
          id = ids[i],
          group=i,
          x=x1[i,],
          s1.exp,
          s2.exp,
          s1.id=s1.ids[i],
          s2.id=s2.ids[i],
          s1.hl = TRUE,
          s2.hl = TRUE,
          peptide = peptide[i]
        )
      )
  }

  row.names(x) <- c(1:dim(x)[1])

  return(x)
}

getMaxPepLevels <-
function(dat){

  pep.levels <- 0
  max.pep.levels <- 0

  for(i in 1:dim(dat)[1]){
    if(overlaps(dat, i)){
        pep.levels <- pep.levels + 1
    }else{
      if(pep.levels > max.pep.levels){
        max.pep.levels <- pep.levels
      }
      pep.levels <- 0
    }
  }

  return(max.pep.levels)
}

getRectangle <-
function(start, stop, level, exp.vals, mod.col){

  x1 <-
    apply(
      array(1:4),
      1,
      FUN = function(x) start + ((stop - start)/4) * x
    )
  x1 <- cbind(start, t(x1))

  x2 <-
    rbind(
      cbind(x1[1], x1[2]),
      cbind(x1[2], x1[3]),
      cbind(x1[3], x1[4]),
      cbind(x1[4], x1[5])
    )
  x2 <- data.frame(x2); names(x2) <- c("xl", "xr")

  #top rectangle (subject 1)
  x3 <- cbind(x2, yb = level)
  x3 <- cbind(x3, yt = level + .15)
```

```
  #bot rectangle (subject 2), duplicate x3, shift lower rec by -.15
  x4 <- rbind(x3, x3)
  x4[5:8, c("yb", "yt")] <-  x4[5:8, c("yb", "yt")] - .15

  x5 <- cbind(x4, exp.vals); names(x5)[5] <- "exp.vals"
  x6 <- cbind(x5, mod.col); names(x6)[6] <- "mod.col"

  return(x6)
}

getSomColors <-
function(som.train, som.proj){

  sp <-
    cbind(
      som.proj,
      xy = paste(som.proj$x, som.proj$y, sep=".")
    )

  st <-
    cbind(
      som.train$code.sum,
      xy = paste(som.train$code.sum$x, som.train$code.sum$y, sep="."),
      id = as.numeric(row.names(som.train$code.sum))
    )

  sp$row.order = row.names(sp)
  ids <- merge(st, sp)
     ids <- ids[order(ids$row.order),][1:6]

  return(ids)
}

getSomTrends <-
function(som.train, trend){

  x <-
    data.frame(
      as.data.frame(som.train$code),
      som.id = c(1:dim(as.data.frame(som.train$code))[1])
    )

  y <- x[eval(expr=trend, envir=x), "som.id"]

  return(y)
}

highlightLines <-
function(x, type, ids){

  if(type == 1){

    x[,"s2.hl"] <- x[,"s1.hl"] <-
      apply(
        x,
        1,
        FUN=function(x) if(x["s1.id"] == x["s2.id"]) T else F
      )

  }else if(type == 2){

    x[,"s1.hl"] <-
      apply(
        x,
        1,
        FUN=function(x) if(any(x["s1.id"] == ids)) T else F
      )
    x[,"s2.hl"] <-
      apply(
        x,
```

```
      1,
      FUN=function(x) if(any(x["s2.id"] == ids)) T else F
    )

  }else if(type == 3){

    x[,"s1.hl"] <-
      apply(
        x,
        1,
        FUN=function(x) if(any(x["id"] == ids$s1)) T else F
      )
    x[,"s2.hl"] <-
      apply(
        x,
        1,
        FUN=function(x) if(any(x["id"] == ids$s2)) T else F
      )
  }

  return(x)
}

isSig <-
function(pvals, cuttoff=0.05){

  sig <- pvals[!is.na(pvals$S1) & pvals$S1 < cuttoff &
               pvals$S2 < cuttoff & pvals$Both > cuttoff,]
  return(sig)
}

lmePeptide <-
function(x){

  ## Set default values
  anova.s1 <- anova.s2 <- anova.all <- data.frame(p=c(NA, NA, NA))

  x$Time <- factor(x$Time)
  x$id <- paste(x$id, x$Subject, sep="_")
  x$id <- factor(x$id)
  x$Subject <- factor(x$Subject)

  ## Create LME Grouping
  x.group <- groupedData(Area ~ Time | id, data=x)

  ## Determine number of observations
  n <- nlevels(factor(x$id))

  ## Perform LME analysis
  if(n >= 4){
    try(lme.all <- lme(Area ~ Time * Subject, random = ~1|id, data=x.group))
    anova.all <- anova(lme.all)
  }

  if(n >= 4){
    try(lme.s1 <- lme(Area ~ Time, random = ~1|id, subset= Subject == 1,
                      data=x.group))

    try(lme.s2 <- lme(Area ~ Time, random = ~1|id, subset= Subject == 2,
                      data=x.group))

    anova.s1 <- anova(lme.s1)
    anova.s2 <- anova(lme.s2)
  }

  ## Create data.frame of p-values
  pvals <-
    data.frame(
      id=x[, "id"][1],
      Peptide=x[, "Peptide"][1],
      Name=x[, "Name"][1],
```

```
      GI=x[, "GI"][1],
      n=n,
      S1=anova.s1$p[2],
      S2=anova.s2$p[2],
      Both=anova.all$p[4]
    )

  return(pvals)
}

LMEPeptide <-
function(idata){

  ## Reshape data, wide to long
  idata <- reshapeData(idata)

  ## LME: call lmePeptide for each level of dat$Peptide
  pvals <- gapply(idata$DATA.LONG, form=Area~Peptide, level="Peptide",
                  FUN=lmePeptide)

  ## Stack pvals, convert to data.frame
  pvals <- as.data.frame(do.call("rbind", pvals))

  ## Order by significance
  pvals.ordered <- pvals[order(pvals$Both),]

  return(pvals.ordered)
}

modCount <-
function(mod, mod.string){
  mval <- regexval(paste("\\(", mod,"\\).*?\\]", sep=""), as.character(mod.string))
  x <- length(strsplit(substring(mval, 5, nchar(mval)-1), ",")[[1]])
  return(x)
}

normLowess <-
function(x){

  x <- LPE::preprocess(x, data.type="MAS5", LOWESS = T)

  # non-log2
  x <- 2^x

  return(x)
}

normMean <-
function(x, trim=0.05){

  for(i in 2:dim(x)[2]){
    x[,i] <- x[, i] * (mean(x[, 1], trim=trim) / mean(x[, i], trim=trim))
  }

  return(x)
}

normMedian <-
function(x){

  for(i in 2:dim(x)[2]){
    x[,i] <- x[, i] * (median(x[, 1]) / median(x[, i]))
  }

  return(x)
}

normMedianSubset <-
function(x, x.subset){

  for(i in 2:dim(x)[2]){
```

```
    x[,i] <- x[, i] * (median(x.subset[, 1]) / median(x.subset[, i]))
  }

  return(x)
}

overlaps <-
function(dat, iq){

  ## Assign start stop values for ith peptide (iq == i)
  s1 <- dat[iq,"Start"]
  s2 <- dat[iq,"Stop"]

  ol <- FALSE
  i <- 1

  ## Check for overlap
  while(!ol & i < dim(dat)[1] & i < iq){

    sp <- dat[i,"Stop"]
    st <- dat[i,"Start"]

    if(s1 <= sp &&  s2 >= st){
      ol <- TRUE
    }else{
      ol <- FALSE
    }

    i <- i + 1
  }

  return(ol)
}

ratio <-
function(x){

  y <- x

  for(i in 1:dim(x)[2]){
    y[,i] <- x[,i] / x[,1]
  }

  return(y)
}

regexval <-
function(pattern, string, perl=T){
  m <- regexpr(pattern, string,perl=T)
  s <- substring(string, m[1], attributes(m)$match.length + m[1]-1)
  return(s)
}

reshapeData <-
function(idata, simple.cols=TRUE){

  dat <- idata$DATA
  dat.length <- dim(dat)[1]
  dat$id <- factor(rownames(dat))

  dat.long <-
    reshape(
      dat,
      idvar="id",
      varying=
        c(
          "Area.113",
          "Area.114",
          "Area.115",
          "Area.116",
          "Area.117",
```

```
          "Area.118",
          "Area.119",
          "Area.121"
        ),
      direction="long"
    )

  ## Add group classifiers
  dat.long$id <- rep(dat[, "id"], 8)

  ## Set Coding Variables: Time and Subject
  for(i in c(1:4)){
    dat.long[grep(idata$TAGS.S1[i], dat.long$time), "Time"] <- i
    dat.long[grep(idata$TAGS.S2[i], dat.long$time), "Time"] <- i

    dat.long[grep(idata$TAGS.S1[i], dat.long$time), "Subject"] <- 1
    dat.long[grep(idata$TAGS.S2[i], dat.long$time), "Subject"] <- 2
  }

  ## Simplify column names
  if(simple.cols){
    names(dat.long)[grep("Best.Peptide.Sequence", names(dat.long))]  <- "Peptide"
    names(dat.long)[grep("Accession.Number", names(dat.long))]  <- "GI"
    names(dat.long)[grep("Protein.Name", names(dat.long))]  <- "Name"
  }

  ## Create groups
  dat.long$Peptide <- factor(dat.long$Peptide)
  dat.long$Time <- factor(dat.long$Time)
  dat.long$id <- factor(dat.long$id)
  dat.long$Subject <- factor(dat.long$Subject)

  idata$DATA.LONG <- dat.long

  return(idata)
}

setEnvironmentVars <-
function(...){

  args <- list(...)

  #create base environment
  #e <- new.env(parent = baseenv())

  for(i in 1:length(args))
    assign(names(args[i]), args[[i]], env=.GlobalEnv)
}

siteCount <-
function(site="K", peptide){
  x <- evalq(if(!is.na(grep("N-term", site)[1])) return(1) else(0))
  m <- gregexpr(site, as.character(peptide), perl=T)[[1]]
  if(m[1] > 0)
      return(length(m)+x)
  else
    return(x)
}

stackSubjects <-
function(dat, cols.s1, cols.s2){

  s1 <-
    data.frame(
      id = paste(
              dat$Accession.Number,
              dat$Start,
              dat$Stop,
              dat$Protein.Name,
              "s1",
              sep="."
```

```
            ),
      dat[,cols.s1]
    )

  s2 <-
    data.frame(
      id = paste(
              dat$Accession.Number,
              dat$Start,
              dat$Stop,
              dat$Protein.Name,
              "s2",
              sep="."
            ),
      dat[,cols.s2]
    )

  names(s1) <- c("id", "t0", "t1", "t2", "t3")
  names(s2) <- c("id", "t0", "t1", "t2", "t3")

  x <- rbind(s1, s2)

  return(x)
}

sumrat <-
function(x){
  y <- x

  for(i in 1:length(x)){
    y[i] <- x[i]/x[1]
  }

  return(y)
}

testSom <-
function(som.train, exp.vals){

  som.proj <- som::som.project(som.train, exp.vals)
  som.colors <- getSomColors(som.train, som.proj)

  return(som.colors)
}

trainSom <-
function(
    idata,
    som.xdim=15,
    som.ydim=12,
    neigh="bubble",
    topol="rect",
    radius=3,
    alpha=.05
  ){

  st.dat <- stackSubjects(idata$DATA, idata$COLS.S1, idata$COLS.S2)

  si <- som.init(st.dat[,2:5], xdim=som.xdim, ydim=som.ydim)
  st <-
    som::som.train(
      st.dat[,2:5],
      code=si,
      xdim=som.xdim,
      ydim=som.ydim,
      alphaType="linear",
      neigh=neigh,
      topol=topol,
      radius=radius,
      alpha=alpha
    )
```

```
  return(st)
}

trimWS <-
function (x)
{
    x <- sub("[ \t\n\r]*$", "", sub("^[ \t\n\r]*", "", x))
    return(x)
}

unlabeledSiteCount <-
function(
        peptide,
        mod.string,
        mod.ids=c("K","Y","N-term")
      ){

    ##Count Possible Modification Sites
    sc <- sum(sapply(mod.ids, function(x) siteCount(x, peptide)))

    ## Count Modifications
    mod.ids <- unique(mod.ids[order(mod.ids)])
    mc <- sum(sapply(mod.ids, function(x) modCount(x, mod.string)))

    ## Calculate siteCount - modCount
    x <- sc-mc

    return(x)
}
```